Multi-View Oriented GPLVM: Expressiveness and Efficiency

Abstract

The multi-view Gaussian process latent variable model (MV-GPLVM) aims to learn a unified representation from multi-view data but is hindered by challenges such as limited kernel expressiveness and low computational efficiency. To overcome these issues, we first introduce a new duality between the spectral density and the kernel function. By modeling the spectral density with a bivariate Gaussian mixture, we then derive a generic and expressive kernel termed Next-Gen Spectral Mixture (NG-SM) for MV-GPLVMs. To address the inherent computational inefficiency of the NG-SM kernel, we propose a random Fourier feature approximation. Combined with a tailored reparameterization trick, this approximation enables scalable variational inference for both the model and the unified latent representations. Numerical evaluations across a diverse range of multi-view datasets demonstrate that our proposed method consistently outperforms state-of-the-art models in learning meaningful latent representations.

1 INTRODUCTION

Multi-view representation learning aims to construct a unified latent representation by integrating multiple modalities and aspects of the observed data [Li et al., 2018, Wang et al., 2015]. The learned representation captures inter-view correlations within observations. By sharing information between each view of the data, we can obtain a much richer latent representation of the data compared to modeling each view independently which is crucial for handling complex datasets [Zhang et al., 2018, Wei et al., 2022, Lu et al., 2019]. For example, modeling video data which involves both visual frames and audio signals [Hussain et al., 2021], or developing clinical diagnostic systems that incorporates the patients' various medical records [Yuan et al., 2018].

Multi-view representation learning first emerged in early works leveraging techniques such as canonical correlation analysis (CCA) [Hotelling, 1992] and its kernelized extensions [Bach and Jordan, 2002, Hardoon et al., 2004]. However, these methods are limited in their ability to capture latent representations in complex datasets. To address this limitation, two more advanced approaches have become standard in multi-view learning: neural network-based methods, exemplified by multi-view variational autoencoders (MV-VAEs) [Wu and Goodman, 2018, Mao et al., 2023], and Gaussian process-based methods, represented by multi-view Gaussian process latent variable models (MV-GPLVMs) [Li et al., 2017, Sun et al., 2020].

MV-VAEs incorporate various view-specific variational autoencoders (VAEs) to address multi-view representation learning [Wu and Goodman, 2018, Mao et al., 2023, Kingma and Welling, 2013, Shi et al., 2019, Xu et al., 2021]. However, they suffer from posterior collapse—an inherent issue in VAEs [Wang et al., 2021]—where the encoder collapses to the prior distribution over the latent variables, thereby failing to capture the underlying structure of the data.

Several hypotheses have been proposed to explain posterior collapse. A well-known contributing factor is the overfitting of the decoder network [Bowman et al., 2016, Sønderby et al., 2016]. One promising direction to mitigate this issue is to introduce regularization in the function space of the decoder, which may help the latent variable model learn more informative representations.

The implicit regularization imposed by the Gaussian process (GP) prior helps prevent MV-GPLVMs from severe overfitting [Li et al., 2024]. However, MV-GPLVMs often lack the kernel flexibility required to model complex representations [Li et al., 2017] and are computationally intensive to train, especially on large-scale multi-view datasets [Sun et al., 2020]. To overcome these challenges, we propose an expressive and efficient multi-view oriented GPLVM. Our contributions are:

- We establish a novel duality between the spectral den-

Submitted to the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025). **To be used for reviewing only**.

sity and the kernel function, deriving the expressive and generic Next-Gen Spectral Mixture (NG-SM) kernel, which, by modeling spectral density as dense Gaussian mixtures, can approximate any continuous kernel with arbitrary precision given enough mixture components. Building on this, we design a novel MV-GPLVM for multi-view scenarios, capable of capturing the unique characteristics of each view, leading to an informative unified latent representation.

- To enhance the computational efficiency, we design a unique unbiased RFF approximation for the NG-SM kernel that is differential w.r.t. kernel hyperparameters. By integrating this RFF approximation with an efficient two-step reparameterization trick, we enable efficient and scalable learning of kernel hyperparameters and unified latent representations within the variational inference framework [Kingma and Welling, 2013], making the proposed model well-suited for multi-view scenarios.
- We validate our model on a range of cross-domain multi-view datasets, including synthetic, image, text, and wireless communication data. The results show that our model consistently outperforms various state-of-the-art (SOTA) MV-VAEs, MV-GPLVMs, and multi-view extensions of SOTA GPLVMs in terms of generating informative unified latent representations.

2 BACKGROUND

Gaussian Processes. Probabilistic models like the Gaussian process latent variable model (GPLVM) [Lawrence, 2005] introduces a regularization effect via a GP-distributed prior that helps prevent overfitting and thus improves generalization from limited samples. A GP $f(\cdot)$ is defined as a real-valued stochastic process defined over the input set $\mathcal{X} \subseteq \mathbb{R}^D$, such that for any finite subset of inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N \subset \mathcal{X}$, the random variables $\mathbf{f} = \{f(\mathbf{x}_n)\}_{n=1}^N$ follow a joint Gaussian distribution [Williams and Rasmussen, 2006]. A common prior choice for a GP-distributed function is:

$$\mathbf{f} \mid \mathbf{X} = \mathcal{N}(\mathbf{f} \mid \mathbf{0}, \mathbf{K}), \tag{1}$$

where $\mathbf{K}$ denotes the covariance/kernel matrix evaluated on the finite input $\mathbf{X}$ with the kernel function $k(\mathbf{x}_1, \mathbf{x}_2)$, i.e., $[\mathbf{K}]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j), i, j \in (1, ..., N)$.

Consequently, the GPLVM has laid the groundwork for several advancements in multi-view representation learning. One of the early works by Li et al. [2017] straightforwardly assumes that each view in observations is a projection from a shared latent space using a GPLVM, referred to as multi-view GPLVM (MV-GPLVM).

Multi-View Gaussian Process Latent Variable Model. A MV-GPLVM assumes that the relationship between each view $v \in (1, ..., V)$ of observed data $\mathbf{Y}^v = [\mathbf{y}^v_{:,1}, \mathbf{y}^v_{:,2}, ..., \mathbf{y}^v_{:,M_v}] \in \mathbb{R}^{N \times M_v}$ and the shared/unified latent variables $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^\top \in \mathbb{R}^{N \times D}$ is modeled by a GP. That is, in each dimension $m \in (1, ..., M_v)$ and view v, MV-GPLVM is defined as follows:

$$\mathbf{y}^v_{:,m} \mid f^v_m(\mathbf{X}) \sim \mathcal{N}(f^v_m(\mathbf{X}), \sigma^2_v \mathbf{I}), \tag{2a}$$

$$f^v_m(\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}^v), \tag{2b}$$

$$\mathbf{x}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2c}$$

where σ^2_v represents the noise variance and $f^v_m(\mathbf{X}) = [f^v_m(\mathbf{x}_1) \ldots f^v_m(\mathbf{x}_N)]^\top \in \mathbb{R}^N$. The stationary radial basis function (RBF) is typically the 'default' choice of the kernel function. Due to the conjugacy between the Gaussian likelihood[1] and the GP, we can integrate out each $f^v_m(\cdot)$ and get the marginal likelihood formed as

$$\mathbf{y}^v_{:,m} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}^v + \sigma^2_v \mathbf{I}). \tag{3}$$

Based on the marginal likelihood and the prior of $\mathbf{X}$ (Eq. (2c)), the *maximum a posteriori* (MAP) estimation of $\mathbf{X}$ can be obtained, with $\mathcal{O}(N^3)$ computational complexity due to the inversion of the kernel matrix. Eleftheriadis et al. [2013] extends the MV-GPLVM with a discriminative prior over the shared latent space for adapted to classification tasks. Later, Sun et al. [2020] incorporates deep GPs [Damianou and Lawrence, 2013] into MV-GPLVM for model flexibility, named MV-DGPLVM. However, existing MV-GPLVMs still fall short in handling practical multi-view datasets, that are often large-scale and exhibit diverse patterns across views. This limitation arises from either (1) the high computational complexity of fitting a deep GP or (2) limited kernel expressiveness caused by the stationary assumption. Particularly, a stationary kernel may fail to model input-varying correlations [Remes et al., 2017], particularly in domains like video analysis and clinical diagnosis that exhibit complex, time-varying dynamics.

To address those issues, we review the recent work of GPLVM [Lalchand et al., 2022, Li et al., 2024, Zhang et al., 2023], that we may extend to deal with the multi-view scenario (See detailed comparisons in Table 1). One potential solution is the *advised*RFLVM (named ARFLVM for short) [Li et al., 2024]. This method integrates the stationary spectral mixture (SM) kernel to enhance kernel flexibility in conjunction with a scalable random Fourier feature (RFF) kernel approximation. However, it remains limited by the stationary assumption.

Random Fourier Features. Bochner's theorem [Bochner et al., 1959] states that any continuous stationary kernel and its spectral density $p(\mathbf{w})$ are Fourier duals, i.e., $k(\mathbf{x}_1 - \mathbf{x}_2) = \int p(\mathbf{w}) \exp\big(i\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2)\big)\, d\mathbf{w}$. Built upon this duality, Rahimi and Recht [2007] approximates the stationary kernel $k(\mathbf{x}_1 - \mathbf{x}_2)$ using an unbiased Monte Carlo

[1] Extending to other likelihoods is straightforward, as guided by the GP literature [Zhang et al., 2023, Lalchand et al., 2022].

Table 1: An overview of relevant MV-LVMs, where we extend advanced GPLVMs to the multi-view case by prefixing them with MV. Additionally, N represents the total number of observations, M denotes the observation dimensions, V refers to the number of views, U refers to the number of inducing points, H indicates the number of GP layer, and L indicates the dimension of random features.

Model	Scalable model	Highly expressive kernel	Probabilistic mapping	Bayesian inference of latent variables	Computational complexity	Reference
MVAE	✓	-	✗	✓	-	Wu and Goodman [2018]
MM-VAE	✓	-	✗	✓	-	Mao et al. [2023], Shi et al. [2019]
MV-GPLVM	✗	✗	✓	✗	$\mathcal{O}(N^3V)$	Zhao et al. [2017]
MV-GPLVM-SVI	✓	✗	✓	✓	$\mathcal{O}(MU^3V)$	Lalchand et al. [2022]
MV-RFLVM	✗	✗	✓	✗	$\mathcal{O}(NL^2V)$	Zhang et al. [2023]
MV-DGPLVM	✗	✗	✓	✓	$\mathcal{O}(HNU^2V)$	Sun et al. [2020]
MV-ARFLVM	✓	✗	✓	✓	$\mathcal{O}(NL^2V)$	Li et al. [2024]
NG-MV-RFLVM	✓	✓	✓	✓	$\mathcal{O}(NL^2V)$	**This work**

(MC) estimator with $L/2$ spectral points $\{\mathbf{w}^{(l)}\}_{l=1}^{L/2}$ sampled from $p(\mathbf{w})$, given by

$$k(\mathbf{x}_1 - \mathbf{x}_2) \approx \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2), \tag{4}$$

where the random feature $\phi(\mathbf{x}) =$

$$\sqrt{\frac{2}{L}}\left[\cos\left(2\pi\mathbf{x}^\top\mathbf{w}^{1:L/2}\right), \sin\left(2\pi\mathbf{x}^\top\mathbf{w}^{1:L/2}\right)\right]^\top \in \mathbb{R}^L.$$

Here the superscript $1 : L/2$ indicates that the cosine or sine function is repeated $L/2$ times, with each element corresponding to the one entry of $\{\mathbf{w}^{(l)}\}_{l=1}^{L/2}$. Consequently, the kernel matrix $\mathbf{K}$ can be approximated by $\Phi(\mathbf{x})\Phi(\mathbf{x})^\top$, with $\Phi(\mathbf{x}) = [\phi(\mathbf{x}_1); \ldots; \phi(\mathbf{x}_N)]^\top$. By employing the RFF approximation, the kernel matrix inversion can be computed using the Woodbury matrix identity [Woodbury, 1950], thereby reducing the computational complexity to $\mathcal{O}(NL^2)$.

3 METHODOLOGY

In § 3.1, we introduce our expressive kernel function for multi-view representation learning. In § 3.2, we propose a sampling-based variational inference algorithm to efficiently learn both kernel hyperparameters and latent representations. The scalable RFF approximation and the associated efficient reparameterization trick are detailed in § 3.3.

3.1 NEXT-GEN SM (NG-SM) KERNEL

As mentioned in § 2, limited kernel expressiveness in the MV-GPLVM may hinder its ability to capture informative latent representations, potentially neglecting crucial view-specific characteristics like time-varying correlations in the data[2]. This problem necessitates a more flexible kernel design.

The bivasriate SM kernel (BSM) is one notable development for improving kernel flexibility [Chen et al., 2024, Remes et al., 2017, Chen et al., 2021]. Grounded in the duality that relates any continuous kernel function to a dual density from the generalized Bochner's theorem [Yaglom, 1987], this method first models the dual density using a mixture of eight bivariate Gaussian components and then transforms it into the BSM kernel. By removing the restrictions of stationarity, the generalized Bochner's theorem enables the BSM kernel to capture time-varying correlations [Remes et al., 2017].

However, the BSM kernel faces several limitations imposed by the generalized Bochner's theorem: (1) To guarantee the positive semi-definite (PSD) spectral density, the two variables in the bivariate Gaussian must have equal variances, which limits the flexibility of the Gaussian mixture and consequently reduces the kernel's expressiveness. (2) According to the duality in the generalized Bochner's theorem, it is not possible to derive a closed form expression of random features[3], such as $\phi(\mathbf{x})$ in Eq. (4). Thus, the inversion of the BSM kernel matrix retains high computational complexity, rendering it unsuitable for multi-view datasets.

To address the limitations of the BSM kernel, we propose the new NG-SM kernel—a more flexible and RFF-admissible alternative. Its formulation is enabled by a novel duality, stated in Theorem 1.

Theorem 1 (Universal Bochner's Theorem)**.** *A complex-valued bounded continuous kernel $k(\mathbf{x}_1, \mathbf{x}_2)$ on $\mathbb{R}^D$ is the covariance function of a mean square continuous complex-valued random process on $\mathbb{R}^D$ if and only if $k(\mathbf{x}_1, \mathbf{x}_2) =$*

$$\frac{1}{4}\int \exp\left(i\mathbf{w}_1^\top\mathbf{x}_1 - i\mathbf{w}_2^\top\mathbf{x}_2\right) + \exp\left(i\mathbf{w}_2^\top\mathbf{x}_1 - i\mathbf{w}_1^\top\mathbf{x}_2\right) + \exp\left(i\mathbf{w}_1^\top\mathbf{x}_1 - i\mathbf{w}_1^\top\mathbf{x}_2\right) + \exp\left(i\mathbf{w}_2^\top\mathbf{x}_1 - i\mathbf{w}_2^\top\mathbf{x}_2\right) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2)$$

where u is the Lebesgue-Stieltjes measure associated with some function $p(\mathbf{w}_1, \mathbf{w}_2)$. When $\mathbf{w}_1 = \mathbf{w}_2$, this theorem reduces to Bochner's theorem.

Proof. The proof can be found in App. B.2. □

The duality established in Theorem 1 implies that a bivariate spectral density entirely determines the properties of a continuous kernel function. In this sense, we approximate the

[2] For an exploration of the impact of limited kernel expressiveness in manifold learning, see § 4.2.

[3] See App. B.1 for a more detailed discussion.

underlying bivariate spectral density by a Gaussian mixture:

$$p_{\text{ngsm}}\left(\mathbf{w}_1,\mathbf{w}_2\right)=\sum_{q=1}^{Q}\alpha_q s_q\left(\mathbf{w}_1,\mathbf{w}_2\right) \tag{5}$$

with each symmetric density $s_q\left(\mathbf{w}_1,\mathbf{w}_2\right)$

$$\begin{aligned} s_q(\mathbf{w}_1,\mathbf{w}_2)=&\frac{1}{2}\mathcal{N}\left(\begin{pmatrix}\mathbf{w}_1\\ \mathbf{w}_2\end{pmatrix}\middle| \begin{pmatrix}\boldsymbol{\mu}_{q1}\\ \boldsymbol{\mu}_{q2}\end{pmatrix},\begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\text{c}}^{\top}\\ \boldsymbol{\Sigma}_{\text{c}} & \boldsymbol{\Sigma}_2\end{bmatrix}\right)+\\ &\frac{1}{2}\mathcal{N}\left(\begin{pmatrix}-\mathbf{w}_1\\ -\mathbf{w}_2\end{pmatrix}\middle| \begin{pmatrix}\boldsymbol{\mu}_{q1}\\ \boldsymbol{\mu}_{q2}\end{pmatrix},\begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\text{c}}^{\top}\\ \boldsymbol{\Sigma}_{\text{c}} & \boldsymbol{\Sigma}_2\end{bmatrix}\right)\end{aligned} \tag{6}$$

in order to explore the space of continuous kernels. To simplify the notation, we omit the index q from the submatrices $\boldsymbol{\Sigma}_1=\operatorname{diag}(\boldsymbol{\sigma}_{q1}^2)$, $\boldsymbol{\Sigma}_2=\operatorname{diag}(\boldsymbol{\sigma}_{q2}^2)$, and $\boldsymbol{\Sigma}_{\text{c}}=\rho_q\operatorname{diag}(\boldsymbol{\sigma}_{q1})\operatorname{diag}(\boldsymbol{\sigma}_{q2})$, where $\boldsymbol{\sigma}_{q1}^2,\boldsymbol{\sigma}_{q2}^2\in\mathbb{R}^D$ and the scalar ρ_q denotes the correlation between $\mathbf{w}_1$ and $\mathbf{w}_2$. These components collectively form the covariance matrix of the q-th bivariate Gaussian component. Furthermore, the vectors $\boldsymbol{\mu}_{q1}$ and $\boldsymbol{\mu}_{q2}\in\mathbb{R}^D$ constitute the mean of the q-th bivariate Gaussian component.

Based on Theorem 1, we derive the NG-SM kernel, $k_{\text{ngsm}}(\mathbf{x}_1,\mathbf{x}_2)$ (see Eq. (39) in App. B.3), with kernel hyperparameters $\boldsymbol{\theta}_{\text{ngsm}}=\{\alpha_q,\boldsymbol{\mu}_{q1},\boldsymbol{\mu}_{q2},\boldsymbol{\sigma}_{q1}^2,\boldsymbol{\sigma}_{q2}^2,\rho_q\}_{q=1}^{Q}$. As the PSD assumption is relaxed and the mixtures of Gaussians become dense [Plataniotis and Hatzinakos, 2000], the duality ensures that the NG-SM kernel can approximate any continuous kernel arbitrary well. Next, we will demonstrate that a general closed-form RFF approximation can be derived for all kernels based on our duality, which we specify for the NG-SM kernel below.

Theorem 2. *Let $\phi(\mathbf{x})$ be the randomized feature map of $k_{ngsm}(\mathbf{x}_1,\mathbf{x}_2)$, defined as follows:*

$$\sqrt{\frac{1}{2L}}\begin{bmatrix}\cos\left(\mathbf{w}_1^{(1:L/2)\top}\mathbf{x}\right)+\cos\left(\mathbf{w}_2^{(1:L/2)\top}\mathbf{x}\right)\\ \sin\left(\mathbf{w}_1^{(1:L/2)\top}\mathbf{x}\right)+\sin\left(\mathbf{w}_2^{(1:L/2)\top}\mathbf{x}\right)\end{bmatrix}\in\mathbb{R}^L, \tag{7}$$

where the vertically stacked vectors $\{[\mathbf{w}_1^{(l)};\mathbf{w}_2^{(l)}]\}_{l=1}^{L/2}\in\mathbb{R}^{L/2\times 2D}$ are independent and identically distributed (i.i.d) random vectors drawn from the spectral density $p_{ngsm}(\mathbf{w}_1,\mathbf{w}_2)$. Then, the unbiased estimator of the kernel $k_{ngsm}(\mathbf{x}_1,\mathbf{x}_2)$ using RFFs is given by:

$$k_{ngsm}(\mathbf{x}_1,\mathbf{x}_2)\approx\phi(\mathbf{x}_1)^{\top}\phi(\mathbf{x}_2). \tag{8}$$

Proof. The proof can be found in App. C.1. □

By integrating this unbiased estimator into the framework of MV-GPLVM, we derive the next-gen multi-view GPLVM:

$$\mathbf{y}_{:,m}^{v}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Phi}_x^{v}\boldsymbol{\Phi}_x^{v\top}+\sigma_v^2\mathbf{I}), \tag{9a}$$

$$(\mathbf{w}_1^{(l)})^{v},(\mathbf{w}_2^{(l)})^{v}\sim p_{\text{ngsm}}^{v}(\mathbf{w}_1^{v},\mathbf{w}_2^{v}), \tag{9b}$$

$$\mathbf{x}_n\sim\mathcal{N}\left(\mathbf{0},\mathbf{I}\right), \tag{9c}$$

where the random feature matrix for each view is denoted as $\boldsymbol{\Phi}_x^{v}=[\phi(\mathbf{x}_1);\ldots;\phi(\mathbf{x}_N)]^{\top}$, with the superscript of each feature map omitted for simplicity of notation. Furthermore, we collectively denote $\boldsymbol{\sigma}^2=\{\sigma_v^2\}_{v=1}^{V}$ and spectral points $\mathbf{W}\triangleq\{\mathbf{W}^{v}\}_{v=1}^{V}$, with each

$$\mathbf{W}^{v}\triangleq\{[(\mathbf{w}_1^{(l)})^{v};(\mathbf{w}_2^{(l)})^{v}]\}_{l=1}^{L/2}\in\mathbb{R}^{L/2\times 2D}. \tag{10}$$

The following subsections will illustrate how to infer the parameters of the NG-MVLVM through a unified sampling-based variational inference framework [Kingma and Welling, 2013].

3.2 SAMPLING-BASED VARIATIONAL INFERENCE

We employ variational inference [Blei et al., 2017] to jointly and efficiently estimate the posterior and hyperparameters $\boldsymbol{\theta}=\{\boldsymbol{\theta}_{\text{ngsm}},\boldsymbol{\sigma}^2\}$. Variational inference reformulates Bayesian inference task as a deterministic optimization problem by approximating the true posterior $p(\mathbf{W},\mathbf{X}|\mathbf{Y})$ using a surrogate distribution, $q(\mathbf{W},\mathbf{X})$, indexed by variational parameters $\boldsymbol{\xi}$. The variational parameters are typically estimated by maximizing the evidence lower bound (ELBO) which is equivalent to minimizing the Kullback-Leibler (KL) divergence $\mathrm{KL}(q(\mathbf{W},\mathbf{X})\|p(\mathbf{W},\mathbf{X}|\mathbf{Y}))$ between the surrogate and the true posterior.

To construct the ELBO for NG-MVLVM, we first obtain joint distribution of the model:

$$p(\mathbf{Y},\mathbf{X},\mathbf{W})=p(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^{v})p(\mathbf{Y}^{v}|\mathbf{X},\mathbf{W}^{v}) \tag{11}$$

and then define the variational distributions $q(\mathbf{W},\mathbf{X})$ as

$$q(\mathbf{X},\mathbf{W})\triangleq q(\mathbf{X})p(\mathbf{W})=q(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^{v}), \tag{12}$$

where $q(\mathbf{X})=\prod_{n=1}^{N}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n,\boldsymbol{S}_n)$, and $\boldsymbol{\xi}\triangleq\{\boldsymbol{\mu}_n\in\mathbb{R}^D,\boldsymbol{S}_n\in\mathbb{R}^{D\times D}\}_{n=1}^{N}$ are the free variational parameters. The variational distribution $q(\mathbf{W})$ is set to its prior $p(\mathbf{W})$ as this is essentially equal to assuming that $q(\mathbf{W})$ follows a Gaussian mixtures (See the justification in App. C.3). Consequently, the optimization problem is:

$$\begin{aligned}\max_{\boldsymbol{\theta},\boldsymbol{\xi}}\ &\mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\log\frac{p(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^{v})p(\mathbf{Y}^{v}|\mathbf{X},\mathbf{W}^{v})}{q(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^{v})}\right]\\ &=\sum_{v=1}^{V}\underbrace{\mathbb{E}_{q(\cdot,\cdot)}\left[\log p(\mathbf{Y}^{v}|\mathbf{X},\mathbf{W}^{v})\right]}_{\text{(a): reconstruction}}-\underbrace{\mathrm{KL}(q(\mathbf{X})\|p(\mathbf{X}))}_{\text{(b): regularization}}\end{aligned}$$

which jointly learns the model hyperparameters $\boldsymbol{\theta}$ and infers the variational parameters $\boldsymbol{\xi}$. Term (a) encourages reconstructing the observations using any samples drawn from $q(\mathbf{X},\mathbf{W})$ and term (b) serves as a regularization to prevent $\mathbf{X}$ from deviating significantly from the prior distribution.

To address this optimization problem using the sampling-based variational inference [Kingma and Welling, 2013], we first derive the analytical form of term (b), the KL divergence between two Gaussian distributions, and expand term (a), approximating the expectation via MC estimation,

$$\sum_{v=1}^{V}\sum_{m=1}^{M_v}\frac{1}{I}\sum_{i=1}^{I}\log\mathcal{N}\left(\mathbf{y}_{:,m}^{v}|\mathbf{0},(\mathbf{\Phi}_x^v\mathbf{\Phi}_x^{v\top})^{(i)}+\sigma_v^2\mathbf{I}\right), \quad (13)$$

where I denotes the number of differentiable MC samples drawn from $q(\mathbf{X})$ and $p(\mathbf{W})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$; See the further computational details in App. C.2. Then, modern optimization techniques, such as Adam [Kingma and Ba, 2014], can be directly applied to solve the problem. However, this raises a question: *How can differentiable MC samples be efficiently generated from the mixture bivariate Gaussian distribution, $p(\mathbf{W}^v)$, that implicitly involves discrete variables?*

3.3 SAMPLING IN MIXTURE BIVARIATE GAUSSIANS

In other words, it is essential to both generate differentiable samples from the mixture bivariate Gaussian and ensure high sampling efficiency, which is particularly beneficial in the multi-view case. However, the typical sampling process hinders us from achieving this goal [Mohamed et al., 2020, Graves, 2016]. A primary difficulty stems from first generating an index q from the discrete distribution, $P(q)=\alpha_q/\sum_{j=1}^{Q}\alpha_j, q=1,...,Q$, as it is inherently non-reparameterizable w.r.t. the mixture weights. Although the Gumbel-Softmax method provides an approximation, it remains unstable and highly sensitive to hyperparameter choices [Potapczynski et al., 2020]. Additionally, directly performing joint sampling from $s_q(\mathbf{w}_1,\mathbf{w}_2)$ (see Eq. (6)) incurs a computational complexity of $\mathcal{O}(8D^3)$, further complicating the sampling process. Next, we will demonstrate how to address both issues. To simplify the notation, we omit the superscript v in this section.

1) Two-step reparameterization trick

Applying the reparameterization trick on a multivariate normal distribution requires computing the Cholesky decomposition of the full covariance matrix, incurring a computational cost of $\mathcal{O}(8D^3)$ [Mohamed et al., 2020]. To alleviate this computational burden, we propose the two-step reparameterization trick as follows, which reducing sampling complexity to $\mathcal{O}(D)$.

Proposition 1 (Two-step reparameterization trick). *We can sample $\mathbf{w}_1^{(l)},\mathbf{w}_2^{(l)}$ from a bivariate Gaussian distribution $s_q(\mathbf{w}_1,\mathbf{w}_2)$ using the following steps:*

1. $\mathbf{w}_1^{(l)}=\boldsymbol{\mu}_{q1}+\boldsymbol{\sigma}_{q1}\circ\boldsymbol{\epsilon}_1,$

2. $\mathbf{w}_2^{(l)}=\boldsymbol{\mu}_{q2}+\rho_q(\boldsymbol{\sigma}_{q2}\backslash\boldsymbol{\sigma}_{q1})\circ(\mathbf{w}_1^{(l)}-\boldsymbol{\mu}_{q1})+\sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2}\circ\boldsymbol{\epsilon}_2,$

Algorithm 1: NG-MVLVM: Next-Gen Multi-View Latent Variable Model

Input: Dataset $\mathbf{Y}$; Maximum iterations T.
Initialize: Iteration count $t=0$; Initialized model hyperparameters $\boldsymbol{\theta}$ and variational parameters $\boldsymbol{\xi}$.
while $t<T$ ***or*** *Not Converged* **do**
- Sample $\mathbf{X}$ from $q(\mathbf{X})=\prod_{n=1}^{N}\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_n,\boldsymbol{S}_n)$ using the reparameterization trick.
- For each view, sample $\mathbf{W}^v$ from $p(\mathbf{W}^v)=\prod_{q=1}^{Q}\prod_{l=1}^{L/2}s_q(\mathbf{w}_1^v,\mathbf{w}_2^v)$ via the two-step reparameterization trick.
- For each view, construct $\mathbf{\Phi}_{\text{ngsm}}^v(\mathbf{X})$ using the sampled $\mathbf{X}$ and $\mathbf{W}^v$.
- Evaluate data reconstruction term of ELBO through Eq. (13).
- Evaluate regularized term of ELBO analytically.
- Maximize ELBO and update $\boldsymbol{\theta}$, $\boldsymbol{\xi}$ using Adam [Kingma and Ba, 2014].
- Increment $t=t+1$.

Output: $\boldsymbol{\theta}$, $\boldsymbol{\xi}$.

where $\boldsymbol{\epsilon}_2$, $\boldsymbol{\epsilon}_1$ are standard normal random variables and $\circ$ and $\backslash$ represents element-wise multiplication and division, respectively.

Proof. The proof and complexity analysis can be found in App. C.4. □

2) Unbiased differential RFF estimator

Let spectral points $\mathbf{W}\triangleq\{[\mathbf{w}_{q1}^{(l)};\mathbf{w}_{q2}^{(l)}]\}_{q=1,l=1}^{Q,L/2}$, be sampled from $p(\mathbf{W})=\prod_{q=1}^{Q}\prod_{l=1}^{L/2}s_q(\mathbf{w}_1,\mathbf{w}_2)$ using the two-step reparameterization trick. Inspired by previous work [Li et al., 2024, Jung et al., 2022], we first use the spectral points from the q-th mixture component to construct the following feature map:

$$\varphi_q(\mathbf{x})\triangleq\sqrt{\alpha_q}\cdot\phi(\mathbf{x};\{\mathbf{w}_{q1}^{(l)}\}_{l=1}^{L/2},\{\mathbf{w}_{q2}^{(l)}\}_{l=1}^{L/2}). \quad (14)$$

Based on the feature maps $\{\varphi_q(\mathbf{x})\}_{q=1}^{Q}$, we can formulate our RFF estimator for the NG-SM kernel as follows.

Theorem 3. *Stacking the feature maps $\varphi_q(\mathbf{x}), q=1,\dots,Q$, yields the final form of the RFF approximation for the NG-SM kernel, $\varphi(\mathbf{x})$:*

$$\varphi(\mathbf{x})=\left[\varphi_1(\mathbf{x})^\top,\varphi_2(\mathbf{x})^\top,\dots,\varphi_m(\mathbf{x})^\top\right]^\top\in\mathbb{R}^{QL}. \quad (15)$$

Built upon this mapping, our unbiased RFF estimator for the NG-SM kernel is reformulated as follows:

$$\mathbb{E}_{p(\mathbf{W})}\left[\varphi(\mathbf{x}_1)^\top\varphi(\mathbf{x}_2)\right]=k_{\text{ngsm}}(\mathbf{x}_1,\mathbf{x}_2). \quad (16)$$

Proof. The proof can be found in App. C.5. □

Table 2: Classification accuracy (%) is evaluated by fitting a KNN classifier ($k = 1$) with five-fold cross-validation. **Mean and standard deviation** are computed over five experiments, and the top performance is in bold.

DATASET	PCA	LDA	ISOMAP	HPF	BGPLVM	GPLVM-SVI	DGPLVM
BRIDGES	84.10 ± 0.76	66.81 ± 5.31	79.77 ± 2.58	54.42 ± 10.96	81.85 ± 3.71	79.61 ± 1.93	64.75 ± 4.80
R-CIFAR	26.78 ± 0.22	22.78 ± 0.66	27.23 ± 0.66	20.80 ± 0.64	27.13 ± 1.46	25.12 ± 1.24	28.16 ± 1.86
R-MNIST	36.56 ± 1.23	23.38 ± 2.69	44.45 ± 2.14	31.49 ± 4.02	56.75 ± 3.37	34.41 ± 5.48	74.82 ± 2.24
NEWSGROUPS	39.24 ± 0.52	39.14 ± 1.89	39.79 ± 1.04	33.44 ± 1.91	38.52 ± 1.05	37.84 ± 1.88	37.63 ± 2.48
YALE	54.37 ± 0.87	33.86 ± 2.38	58.84 ± 1.72	51.17 ± 1.96	55.35 ± 3.65	52.17 ± 1.59	76.06 ± 2.60
DATASET	VAE	NBVAE	DCA	CVQ-VAE	RFLVM	ARFLVM	**OURS**
BRIDGES	75.15 ± 1.63	75.85 ± 3.88	70.21 ± 3.63	68.86 ± 1.38	84.61 ± 3.95	84.64 ± 1.54	**85.30 ± 1.27**
R-CIFAR	26.65 ± 0.25	25.97 ± 0.50	25.51 ± 1.90	22.45 ± 1.23	28.47 ± 10.34	29.02 ± 0.62	**31.44 ± 0.68**
R-MNIST	64.37 ± 2.16	28.18 ± 1.20	17.19 ± 7.54	12.87 ± 0.53	60.20 ± 5.53	79.59 ± 1.52	**80.99 ± 0.59**
NEWSGROUPS	38.52 ± 0.29	39.87 ± 1.00	39.97 ± 3.41	35.60 ± 1.96	41.35 ± 0.95	**41.82 ± 0.75**	40.10 ± 1.48
YALE	61.16 ± 2.04	45.60 ± 4.68	28.49 ± 5.40	33.89 ± 0.28	65.37 ± 6.79	76.56 ± 1.02	**76.60 ± 1.96**

Moreover, given inputs $\mathbf{X}$ and the RFF feature map $\varphi(\mathbf{x})$, we can establish the approximation error bound for the NG-SM kernel gram matrix approximation, $\hat{\mathbf{K}}_{\text{ngsm}} \triangleq \Phi_{\text{ngsm}}(\mathbf{X})\Phi_{\text{ngsm}}(\mathbf{X})^\top$ below, where $\Phi_{\text{ngsm}}(\mathbf{X}) = \left[\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_N)\right]^\top \in \mathbb{R}^{N\times QL}$.

Theorem 4. *Let $C = (\sum_{q=1}^{Q} \alpha_q^2)^{1/2}$, then for a small $\epsilon > 0$, the approximation error between the true NG-SM kernel matrix $\mathbf{K}_{\text{ngsm}}$ and its RFF approximation $\hat{\mathbf{K}}_{\text{ngsm}}$ is bounded as follows:*

$$P\left(\left\|\hat{\mathbf{K}}_{\text{ngsm}} - \mathbf{K}_{\text{ngsm}}\right\|_2 \geq \epsilon\right) \leq N \exp\left(\frac{-3\epsilon^2 L}{2NC\left(6\left\|\mathbf{K}_{\text{ngsm}}\right\|_2 + 3NC\sqrt{Q} + 8\epsilon\right)}\right),$$

where $\|\cdot\|_2$ is the matrix spectral norm.

Proof. The proof can be found in App. C.6. □

The feature map $\varphi(\mathbf{x})$ not only provides theoretical guarantees for the approximation but also eliminates the need to generate differential samples from the mixture bivariate Gaussian distribution by directly introducing differentiability into the optimization objective w.r.t. the mixture weights α_q.

Furthermore, the two-step reparameterization trick uses the correlations between $\mathbf{w}_1$ and $\mathbf{w}_2$, for efficient sampling. Consequently, the aforementioned optimization problem is solvable with standard algorithms like Adam [Kingma and Welling, 2013]. We summarize the pseudocode in Algorithm 1 which illustrates the implementation of our proposed method, named Next-Gen SM kernel-embedded MV Random Feature-based GPLVM (NG-MV-RFLVM).

4 EXPERIMENTS

In § 4.1, we demonstrate the capability of our method in representation learning using various single-view real-world datasets. Subsequently, we validate the superior performance of our model on multiple cross-domain multi-view datasets, including synthetic data (§ 4.2), image and text data (§ 4.3), and wireless communication data (§ 4.4). Further experimental details are provided in App. D, with specific focus on benchmark implementations in App. D.2 and the hyperparameter selection process in App. D.3. Additionally, we present supplementary simulation results to highlight the scalability and robustness of our method in handling a large number of views (App. D.4) and missing data imputation (App. D.5).

4.1 SINGLE-VIEW DATA

In this section, we first examine the following single-view data types: images (R-MNIST, YALE, R-CIFAR), text (NEWSGROUPS), and structured data (BRIDGES). To accommodate the high computational costs of RFLVM, the dataset sizes for CIFAR and MNIST are reduced and denoted with the prefix 'R-' (see details in Appendix D.1.2).

The results[4] in Table 2 present the mean and standard deviation of classification accuracy from a five-fold cross-validated K-nearest neighbor (KNN) classification, performed on the learned latent variables for each dataset and method.

Based on this experiment, our method is capable of capturing informative latent representations across various datasets due to its superior modeling and learning capacities. The inferior performance of classic approaches is primarily due to their insufficient learning capacity. The GPLVM variants perform slightly inferior to our approach, primarily due to the assumption of kernel stationarity.

While DGPLVM addresses the modeling limitation by incorporating deep structures, its performance is hindered by more complex model optimization processes [Dunlop et al.,

[4]Comparison benchmarks include various classic dimensionality reduction approaches [Wold et al., 1987, Gopalan et al., 2015, Blei et al., 2003, Bach and Jordan, 2002], GPLVM-based approaches [Lalchand et al., 2022, Li et al., 2024, Zhang et al., 2023, Titsias and Lawrence, 2010], and VAE-based models [Kingma and Welling, 2013, Zhao et al., 2020, Eraslan et al., 2019, Zheng and Vedaldi, 2023]. See further details in Table 7.

Table 3: Classification accuracy, expressed as a percentage (%), is evaluated by using KNN and SVM classifiers with five-fold cross-validation. **Mean of the accuracy** is computed over five experiments and the top-performing result for each metric is in bold.

DATASET	BRIDGES		CIFAR		MNIST		NEWGROUPS		YALE	
METRIC	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM	KNN	SVM
OURS	**85.31 ± 1.47**	**87.49 ± 1.76**	**36.02 ± 0.34**	32.47 ± 0.96	**82.72 ± 0.61**	**60.14 ± 1.11**	**100.0 ± 0.00**	**100.0 ± 0.00**	**83.87 ± 1.44**	**64.81 ± 2.50**
MV-ARFLVM	83.42 ± 1.53	86.21 ± 7.59	35.71 ± 0.59	31.58 ± 1.13	81.85 ± 1.58	59.51 ± 2.43	**100.0 ± 0.00**	**100.0 ± 0.00**	78.86 ± 2.23	56.72 ± 3.41
MV-DGPLVM	78.03 ± 11.96	73.48 ± 3.08	22.15 ± 10.54	24.31 ± 8.07	26.83 ± 12.25	27.04 ± 18.25	60.14 ± 2.31	36.02 ± 5.354	53.97 ± 6.37	34.58 ± 12.67
MVAE	73.38 ± 2.83	71.26 ± 3.22	25.83 ± 7.97	31.74 ± 2.73	46.41 ± 0.88	55.18 ± 1.91	33.81 ± 4.18	35.97 ± 3.25	74.92 ± 3.31	65.13 ± 2.34
MMVAE	73.01 ± 3.01	72.04 ± 2.43	29.81 ± 2.24	**34.73 ± 1.64**	39.07 ± 2.58	40.85 ± 7.90	95.21 ± 1.98	38.86 ± 2.70	41.35 ± 4.94	22.12 ± 5.97
MV-NGPLVM	80.72 ± 3.89	71.58 ± 3.59	29.14 ± 2.07	21.92 ± 1.72	52.74 ± 5.10	13.63 ± 1.96	99.25 ± 4.84	35.28 ± 1.41	30.37 ± 1.36	23.91 ± 4.09

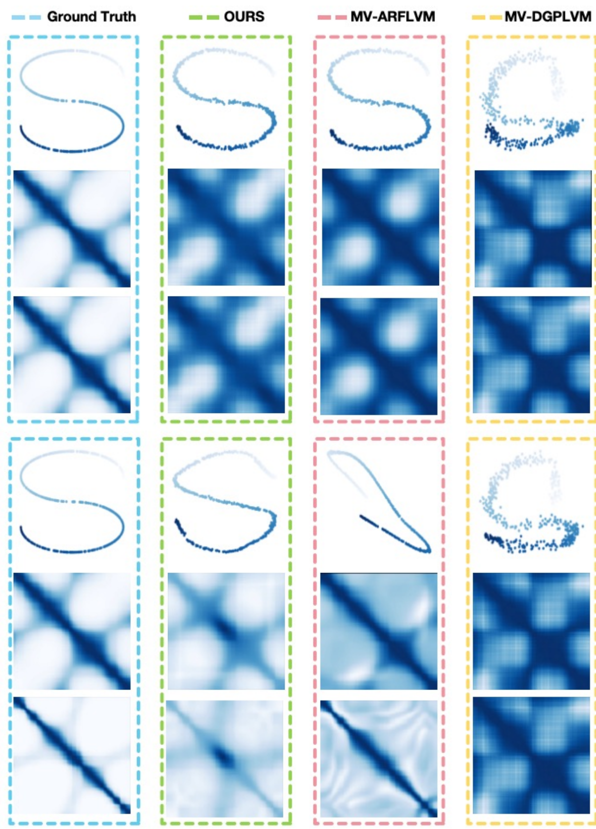


Figure 1: Comparison of learned latent variables and kernel matrices with the ground truth for two-view datasets. Each dashed box contains three rows: the first row shows the latent variables, while the second and third rows represent the kernel matrices for the two views. **Top**: Both views are generated using an RBF kernel. **Bottom**: The first view is generated using an RBF kernel, and the second view is generated using a Gibbs kernel.

2018]. Similarly, VAE-based methods inherently suffer from posterior collapse, making them prone to generating uninformative latent representations, partly due to overfitting [Sønderby et al., 2016, Bowman et al., 2016].

4.2 SYNTHETIC DATA

We further demonstrate the impact of kernel expressiveness on manifold learning and highlight the expressive power of the proposed NG-SM kernel. To this end, we synthesized two datasets using a two-view MV-GPLVM with the S-shape $\mathbf{X}$, based on different kernel configurations: (1) both views using the stationary RBF kernel, and (2) one view using the RBF kernel and the other using the non-stationary Gibbs kernel [Williams and Rasmussen, 2006]. See more details in App. D.1.1.. For benchmark methods, we use the MV-DGPLVM [Sun et al., 2020] and the multi-view extension of the SOTA ARFLVM [Li et al., 2024], namely, MV-ARFLVM.

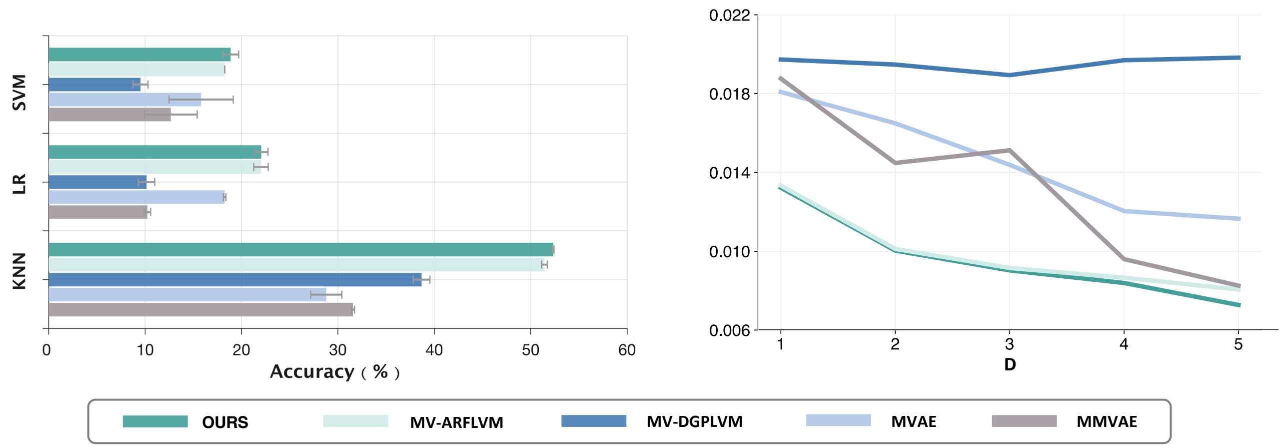
The manifold learning and kernel learning results across different methods for the two datasets are presented in Figure 1. The results for MV-ARFLVM indicate that if the model fails to capture the non-stationary features of one view, then significant distortions arise in the unified latent variables. In turn this degrades the model's ability to learn stationary features from other views. In contrast, both the latent variables and kernel matrices learned by our method are closer to the ground truth compared to the benchmark methods, especially in non-stationary kernel setting. Thus, these results demonstrate the importance of kernel flexibility for the MV-GPLVM.

For the MV-DGPLVM, we select the best latent representations from all GP layers and plot the corresponding kernel Gram matrix. However, the kernel learning result may be less meaningful, as the flexibility of model stems from using deep GPs rather than the kernel choice. The deep architecture of MV-DGPLVM exhibits a notable capability to capture non-stationary components, yielding a relatively "high-quality" latent representations. However, due to the large number of parameters and the complex hierarchical structure [Dunlop et al., 2018], MV-DGPLVM often suffers from overfitting and computational inefficiency, still resulting in inferior performance compared to our method.

4.3 MULTI-VIEW IMAGE AND TEXT DATA

We further illustrate the capability of our model to capture unified latent representations in various multi-view image and text datasets, while maintaining high computational efficiency. Specifically, we follow the setup of Wu and Goodman [2018] and treat the single-view data in § 4.1 and its label as two distinct views. We compare the SOTA MV-GPLVM variants: MV-ARFLVM, MV-DGPLVM, MV-GPLVM with the Gibbs kernel (MV-NGPLVM) and the SOTA MV-VAE variants: MVAE [Wu and Goodman, 2018] and MMVAE [Mao et al., 2023]. After inferring the uni-

Figure 2: (**Left**) Classification accuracy , expressed as a percentage (%), is evaluated using KNN, Logistic Regression (LR) and SVM classifiers with five-fold cross-validation. Mean and standard deviation are computed over five experiments. (**Right**) The average mean squared error (MSE) of the reconstructed channel data compared to the original channel data across different latent variable dimensions in five experiments.



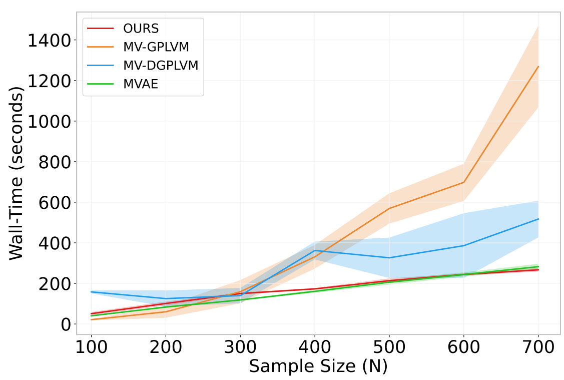


Figure 3: The wall-time for model learning on MNIST across various benchmark methods w.r.t. dataset size N. Solid lines show the average over five runs; shaded areas denote the standard deviation.

fied latent variable $\mathbf{X}$, we perform five-fold cross-validation using two types of classifiers: KNN and support vector machine (SVM). As summarized in Table 3, we report the mean classification accuracy of various classifiers across multiple datasets. Additionally, Figure 3 illustrates the model fitting wall-time on the MNIST dataset with respect to dataset size.

From those results, we can see that our method demonstrates superior performance in estimating the latent represntations while maintaining computational scalability. For the MV-VAE variants, the inferior performance can be attributed to: (1) optimizing the huge number of neural network parameters leads to model instability, and (2) the posterior collapse issue in the VAE results in an uninformative latent space. Both factors are partly due to overfitting, a problem that can be naturally addressed by the MV-GPLVM variants.

Consequently, the performance of the MV-GPLVM variants is generally superior to that of the MV-VAE, though this often comes with increased computational cost—an issue our method effectively mitigates. Specifically, the performance improvement of our method compared to MV-ARFLVM and MV-NGPLVM is due to our proposed NG-SM kernel. MV-DGPLVM not only exhibits instability but also incurs extremely high computational costs, which can be attributed to its large number of parameters.

4.4 WIRELESS COMMUNICATION DATA

For further evaluations, we test the model in a channel compression task. Specifically, we generate wireless communication channel datasets using a high-fidelity channel simulator—QUAsi Deterministic RadIo channel GenerAtor (QUADRIGA)[5]. The environment we considered consists of a base station (BS) with 32 antennas serving 10 single-antenna user equipments (UEs), each moving at a speed of 30 km/h. We sample $1,000$ complex-valued channel vectors for each UE at intervals of 2.5 ms. The real and imaginary parts of each complex channel vector are treated as two distinct views, with the identifier of the UE as the label, resulting in a two-view dataset[6] with $N = 10,000$ and $M_v = 32$ for $v = 1, 2$.

To reduce communication overhead, the UE typically transmits a compressed channel vector to the BS instead of the full vector. This compressed vector must retain sufficient information to accurately reconstruct the ground-truth channel vectors. We use the unified latent variables as the compressed channel vector and evaluate both their classification performance and reconstruction capability. The mean and standard error of the classification accuracy, along with the mean square error (MSE) between the reconstructed and ground-truth channel vectors, are shown in Figure 2. The results indicate that the KNN accuracy achieved by our method consistently surpasses that of competing methods, while the MSE is consistently lower. These findings high-

[5]https://quadriga-channel-model.de

[6]See App. D.1.3 for more details regarding the wireless communication scenario settings.

light the superior performance of our model in channel compression tasks compared to other benchmark approaches.

5 CONCLUSION

This paper introduces NG-MV-RFLVM, a novel model designed to address two critical challenges in multi-view representation learning with MV-GPLVMs: limited kernel expressiveness and computational inefficiency. Specifically, we first establish a duality between spectral density and kernel function, yielding a versatile kernel capable of modeling the non-stationarity in multi-view datasets. We then show that with the proposed RFF approximation and efficient sampling method, the inference of model and latent representations can be effectively performed within a variational inference framework. Experimental validations demonstrate that our model, NG-MV-RFLVM, outperforms state-of-the-art methods such as MV-VAE and MV-GPLVM in providing informative unified latent representations across diverse cross-domain multi-view datasets.

6 FUTURE WORK

While the proposed NG-MVLVM framework effectively captures informative unified latent representations, there remains room for further enhancement. In particular, the current model does not explicitly account for cross-view dependencies, which may limit its performance in scenarios with structured dependencies across views. As a future direction, we plan to incorporate cross-view interaction terms to further enhance the model's capacity for multi-view representation learning. Additionally, we aim to apply our model to real-world multi-modal and medical time-series data, where modeling complex non-stationary behaviors and cross-modal relationships is critical.

References

Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

Mukund Balasubramanian and Eric L Schwartz. The Isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Salomon Bochner et al. *Lectures on Fourier integrals*, volume 42. Princeton University Press, 1959.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *SIGNLL Conference on Computational Natural Language Learning*, 2016.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: Experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Kai Chen, Twan van Laarhoven, and Elena Marchiori. Gaussian processes with skewed Laplace spectral mixture kernels for long-term forecasting. *Machine Learning*, 110:2213–2238, 2021.

Kai Chen, Twan van Laarhoven, and Elena Marchiori. Compressing spectral kernels in Gaussian process: Enhanced generalization and interpretability. *Pattern Recognition*, page 110642, 2024.

AC Damianou and ND Lawrence. Deep Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, 2013.

AC Damianou, Carl Henrik Ek, MK Titsias, and ND Lawrence. Manifold relevance determination. In *International Conference on Machine Learning*, pages 145–152, 2012.

Andreas Damianou, Neil D Lawrence, and Carl Henrik Ek. Multi-view learning as a nonparametric nonlinear inter-battery factor analysis. *Journal of Machine Learning Research*, 22(86):1–51, 2021.

Matthew M Dunlop, Mark A Girolami, Andrew M Stuart, and Aretha L Teckentrup. How deep are deep Gaussian processes? *Journal of Machine Learning Research*, 19 (54):1–46, 2018.

Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Shared Gaussian process latent variable model for multi-view facial expression recognition. In *International Symposium on Visual Computing*, pages 527–538. Springer, 2013.

Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10(1):390, 2019.

Prem Gopalan, Jake M Hofman, and David M Blei. Scalable recommendation with hierarchical Poisson factorization. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

Alex Graves. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.

Zhibin Gu, Songhe Feng, Zhendong Li, Jiazheng Yuan, and Jun Liu. Noodle: Joint cross-view discrepancy discovery and high-order correlation detection for multi-view subspace clustering. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–23, 2024a.

Zhibin Gu, Zhendong Li, and Songhe Feng. Topology-driven multi-view clustering via tensorial refined sigmoid rank minimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 920–931, 2024b.

Gregory Gundersen, Michael Zhang, and Barbara Engelhardt. Latent variable modeling with random features. In *International Conference on Artificial Intelligence and Statistics*, pages 1333–1341. PMLR, 2021.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12): 2639–2664, 2004.

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 162–190. Springer, 1992.

Tanveer Hussain, Khan Muhammad, Weiping Ding, Jaime Lloret, Sung Wook Baik, and Victor Hugo C De Albuquerque. A comprehensive survey of multi-view video summarization. *Pattern Recognition*, 109:107567, 2021.

HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Multi-view representation learning via total correlation objective. In *Advances in Neural Information Processing Systems*, volume 34, pages 12194–12207, 2021.

Yohan Jung, Kyungwoo Song, and Jinkyoo Park. Efficient approximate inference for stationary kernel on frequency domain. In *International Conference on Machine Learning*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Vidhi Lalchand, Aditya Ravuri, and Neil D. Lawrence. Generalised GPLVM with stochastic variational inference. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6(11), 2005.

Jinxing Li, Bob Zhang, and David Zhang. Shared autoencoder Gaussian process latent variable model for visual classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4272–4286, 2017.

Ying Li, Zhidi Lin, Feng Yin, and Michael Minyi Zhang. Preventing model collapse in Gaussian process latent variable models. In *International Conference on Machine Learning*, 2024.

Yingming Li, Ming Yang, and Zhongfei Zhang. A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1863–1883, 2018.

Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Yuxin Mao, Jing Zhang, Mochu Xiang, Yiran Zhong, and Yuchao Dai. Multimodal variational auto-encoder based audio-visual segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 954–965, 2023.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Emanuele Palumbo, Imant Daunhawer, and Julia E Vogt. Mmvae+: Enhancing the generative quality of multimodal vaes without compromises. In *The Eleventh International Conference on Learning Representations*. OpenReview, 2023.

Kostantinos N Plataniotis and Dimitris Hatzinakos. Gaussian mixtures and their applications to signal processing. *Advanced Signal Processing Handbook*, pages 89–124, 2000.

Andres Potapczynski, Gabriel Loaiza-Ganem, and John P Cunningham. Invertible Gaussian reparameterization: Revisiting the Gumbel-softmax. *Advances in Neural Information Processing Systems*, 33:12311–12321, 2020.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 2007.

Sami Remes, Markus Heinonen, and Samuel Kaski. Non-stationary spectral kernels. In *Advances in Neural Information Processing Systems*, 2017.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. *Advances in neural information processing systems*, 30, 2017.

Yves-Laurent Kom Samo and Stephen Roberts. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236*, 2015.

Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In *Advances in Neural Information Processing Systems*, 2019.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, 2016.

Shiliang Sun, Wenbo Dong, and Qiuyang Liu. Multi-view representation learning with deep Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4453–4468, 2020.

Thomas Sutter, Imant Daunhawer, and Julia Vogt. Multimodal generative learning utilizing Jensen-Shannon-divergence. In *Advances in Neural Information Processing Systems*, 2020.

Michalis Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Jean-Francois Ton, Seth Flaxman, Dino Sejdinovic, and Samir Bhatt. Spatial mapping with Gaussian processes and nonstationary Fourier features. *Spatial Statistics*, 28: 59–78, 2018.

Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, 2015.

Yixin Wang, David Blei, and John P Cunningham. Posterior collapse and latent variable non-identifiability. *Advances in Neural Information Processing Systems*, 2021.

Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *European Conference on Computer Vision*, 2022.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, 2013.

Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

Max A Woodbury. *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.

Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, 2018.

Jie Xu, Yazhou Ren, Huayi Tang, Xiaorong Pu, Xiaofeng Zhu, Ming Zeng, and Lifang He. Multi-VAE: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9234–9243, 2021.

Akiva M Yaglom. *Correlation Theory of Stationary and Related Random Functions, Volume I: Basic Results*. Springer, 1987.

Ye Yuan, Guangxu Xun, Kebin Jia, and Aidong Zhang. A multi-view deep learning framework for EEG seizure detection. *IEEE Journal of Biomedical and Health Informatics*, 23(1):83–94, 2018.

Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. Generalized latent multi-view subspace clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(1):86–99, 2018.

Michael Minyi Zhang, Gregory W Gundersen, and Barbara E Engelhardt. Bayesian non-linear latent variable modeling via random Fourier features. *arXiv preprint arXiv:2306.08352*, 2023.

He Zhao, Piyush Rai, Lan Du, Wray Buntine, Dinh Phung, and Mingyuan Zhou. Variational autoencoders for sparse and overdispersed discrete data. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38:43–54, 2017.

Chuanxia Zheng and Andrea Vedaldi. Online clustered codebook. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Multi-View Oriented GPLVM: Expressiveness and Efficiency (Supplementary Material)

APPENDICES

A NOTATION TABLE

We summarize key symbols and their corresponding dimensions used throughout the paper to improve clarity.

Table 4: Summary of Notations

Symbol	Description	Dimension
N	Number of observations	scalar
M	Dimensionality of observation	scalar
V	Number of views	scalar
D	Dimensionality of latent space	scalar
U	Number of inducing points	scalar
H	Number of GP layers (for DGPLVM)	scalar
L	Dimensionality of random features	scalar
X	Latent representation	$N \times D$
$Y^{(v)}$	Observation in view v	$N \times M_v$
$\mathbf{y}^v_{:,m}$	Observation vector in view v, dimension m	$N \times 1$
$f^{(v)}_m(X)$	Latent function for view v, dimension m	$N \times 1$
$K^{(v)}$	Kernel matrix for view v	$N \times N$
$\mathbf{w}^{(v)}$	Spectral points of view v	$L/2 \times 2D$
$(\mathbf{w}^{(l)}_1)^v$	l-th sampled spectral point for view v (first component)	D
σ^2_v	Noise variance for view v	scalar
α_q	Mixture weight of q-th Gaussian component	scalar
μ_{q1}, μ_{q2}	Mean vectors of q-th bivariate Gaussian	D
$\sigma^2_{q1}, \sigma^2_{q2}$	Variance vectors of q-th bivariate Gaussian	D
ρ_q	Correlation coefficient in q-th component	scalar
θ	Set of kernel hyperparameters	-
ξ	Set of variational parameters	-

B NEXT-GEN SM KERNEL AND UNIVERSAL BOCHNER'S THEOREM

B.1 BIVARIATE SM KERNEL

The development of the spectral mixture (SM) kernel is based on the fundamental Bochner's theorem [Wilson and Adams, 2013], which suggests that any stationary kernel and its spectral density are Fourier duals. However, the stationarity assumption limits the kernel's learning capacity, especially when dealing with multi-view datasets, where some views may exhibit non-stationary characteristics. To model the non-stationarity, the Bivariate SM (BSM) kernel was introduced in [Remes et al., 2017, Samo and Roberts, 2015], based on the following generalized Bochner's theorem:

Theorem 5 (Generalized Bochner's Theorem)**.** *A complex-valued bounded continuous kernel $k(\mathbf{x}_1, \mathbf{x}_2)$ on $\mathbb{R}^D$ is the covariance function of a mean square continuous complex-valued random process on $\mathbb{R}^D$ if and only if*

$$k(\mathbf{x}_1, \mathbf{x}_2) = \int \exp\big(i \left(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2\right)\big) m(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2), \tag{17}$$

where m is the Lebesgue-Stieltjes measure associated with some positive semi-definite (PSD) function $S(\mathbf{w}_1, \mathbf{w}_2)$.

According to Theorem 5, one can approximate the function $S(\mathbf{w}_1, \mathbf{w}_2)$, and implement the inverse Fourier transform shown in Eq. (17) to obtain the associated kernel function. The bivariate spectral mixture (BSM) kernel adopts this concept and approximates $S(\mathbf{w}_1, \mathbf{w}_2)$ as a bivariate Gaussian mixture as follows:

$$S(\mathbf{w}_1, \mathbf{w}_2) = \sum_{q=1}^{Q} \alpha_q s_q(\mathbf{w}_{q1}, \mathbf{w}_{q2}), \tag{18}$$

where s_q represents the q-th component of a bivariate Gaussian distribution, formally defined as [Remes et al., 2017]:

$$s_q\left(\mathbf{w}_{q1}, \mathbf{w}_{q2}\right) = \frac{1}{8} \sum_{\boldsymbol{\mu}_q \in \pm\{\boldsymbol{\mu}_{q1}, \boldsymbol{\mu}_{q2}\}^2} \mathcal{N}\left[\begin{pmatrix}\mathbf{w}_{q1}\\ \mathbf{w}_{q2}\end{pmatrix} \middle| \begin{pmatrix}\boldsymbol{\mu}_{q1}\\ \boldsymbol{\mu}_{q2}\end{pmatrix}, \underbrace{\begin{pmatrix}\operatorname{diag}(\boldsymbol{\sigma}_{q1}^2) & \rho_q \operatorname{diag}(\boldsymbol{\sigma}_{q1}) \operatorname{diag}(\boldsymbol{\sigma}_{q2})\\ \rho_q \operatorname{diag}(\boldsymbol{\sigma}_{q1}) \operatorname{diag}(\boldsymbol{\sigma}_{q2}) & \operatorname{diag}(\boldsymbol{\sigma}_{q2}^2)\end{pmatrix}}_{\triangleq \boldsymbol{\Sigma}_q}\right]. \tag{19}$$

Here $\pm\{\boldsymbol{\mu}_{q1}, \boldsymbol{\mu}_{q2}\}^2$ represents:

$$\{(\boldsymbol{\mu}_{q1}, \boldsymbol{\mu}_{q2}), (\boldsymbol{\mu}_{q1}, \boldsymbol{\mu}_{q1}), (\boldsymbol{\mu}_{q2}, \boldsymbol{\mu}_{q2}), (\boldsymbol{\mu}_{q2}, \boldsymbol{\mu}_{q1}), (-\boldsymbol{\mu}_{q1}, -\boldsymbol{\mu}_{q2}), (-\boldsymbol{\mu}_{q1}, -\boldsymbol{\mu}_{q1}), (-\boldsymbol{\mu}_{q2}, -\boldsymbol{\mu}_{q2}), (-\boldsymbol{\mu}_{q2}, -\boldsymbol{\mu}_{q1})\}.$$

The terms $\boldsymbol{\sigma}_{q1}^2$ and $\boldsymbol{\sigma}_{q2}^2 \in \mathbb{R}^D$ represent the variances of the q-th bivariate Gaussian component, while ρ_q denotes the correlation between $\mathbf{w}_{q1}$ and $\mathbf{w}_{q2}$; the vectors $\boldsymbol{\mu}_{q1}$ and $\boldsymbol{\mu}_{q2} \in \mathbb{R}^D$ specify the means of the q-th Gaussian component. The mixture weight is denoted by α_q, and Q is the total number of mixture components. Taking the inverse Fourier transform, we can obtain the following kernel function:

$$k\left(\mathbf{x}_1, \mathbf{x}_2\right) = \sum_{q=1}^{Q} \alpha_q \exp\left(-\frac{1}{2} \tilde{\boldsymbol{x}}^\top \boldsymbol{\Sigma}_q \tilde{\boldsymbol{x}}\right) \Psi_q(\mathbf{x}_1)^\top \Psi_q\left(\mathbf{x}_2\right), \tag{20}$$

where

$$\Psi_q(\mathbf{x}) = \begin{bmatrix}\cos\left(\boldsymbol{\mu}_{q1}^\top \mathbf{x}\right) + \cos\left(\boldsymbol{\mu}_{q2}^\top \mathbf{x}\right)\\ \sin\left(\boldsymbol{\mu}_{q1}^\top \mathbf{x}\right) + \sin\left(\boldsymbol{\mu}_{q2}^\top \mathbf{x}\right)\end{bmatrix} \in \mathbb{R}^2,$$

and $\tilde{\mathbf{x}} = (\mathbf{x}_1, -\mathbf{x}_2) \in \mathbb{R}^{2D}$ is a stacked vector.

The BSM kernel overcomes the stationarity limitation in the SM kernel. However, it still has two major issues.

- First, the assumption of identical variance of $\mathbf{w}_1$ and $\mathbf{w}_2$ limits the approximation flexibility of the mixture of Gaussian, which in turn diminishes the generalization capacity of the kernel.
- Second, the RFF kernel approximation technique cannot be directly applied, as the closed-form expression of the feature map is hard to derive (see explaination below).

The first limitation arises from the requirement that the spectral density must be PSD. To ensure the symmetry of the BSM kernel function (i.e., $k(\mathbf{x}_1, \mathbf{x}_2) = k(\mathbf{x}_2, \mathbf{x}_1)$), the BSM kernel assumes that $\boldsymbol{\sigma}_{q1} = \boldsymbol{\sigma}_{q2}$, $\forall q$.

In addition, we highlight the challenges of deriving a closed-form feature map for RFF when directly applying Theorem 5. According to Theorem 5, the kernel can be approximated via MC as follows:

$$\begin{aligned}
k\left(\mathbf{x}_1, \mathbf{x}_2\right) &= \frac{1}{4} \int \exp\left(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\right) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2)\\
&= \frac{1}{4} \mathbb{E}_u \left(\exp\left(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\right)\right)\\
&\approx \frac{1}{2L} \sum_{l=1}^{L/2} \exp\left(i(\mathbf{w}_1^{(l)\top} \mathbf{x}_1 - \mathbf{w}_2^{(l)\top} \mathbf{x}_2)\right)\\
&= \frac{1}{2L} \sum_{l=1}^{L/2} \left(\cos\left(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\right) \cos\left(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\right) + \sin\left(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\right) \sin\left(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\right)\right).
\end{aligned} \tag{21}$$

Since $\mathbf{w}_1$ and $\mathbf{w}_2$ are distinct, it is hard to derive a closed-form RFF approximation for $k(\mathbf{x}_1, \mathbf{x}_2)$. More specifically, it is challenging to explicitly define $\phi_{\mathbf{w}_1}(\cdot), \phi_{\mathbf{w}_2}(\cdot)$, the feature maps of the kernel function, such that

$$k(\mathbf{x}_1, \mathbf{x}_2) \approx \phi_{\mathbf{w}_1}(\mathbf{x}_1)^\top \phi_{\mathbf{w}_2}(\mathbf{x}_2) = \phi_{\mathbf{w}_1}(\mathbf{x}_2)^\top \phi_{\mathbf{w}_2}(\mathbf{x}_1) \tag{22}$$

Thus, the inversion of the BSM kernel matrix retains high computational complexity, rendering it unsuitable for multi-view data.

Remark 1. *To enhance the kernel capacity, this paper proposes the Universal Bochner's Theorem (Theorem 1) and the NG-SM kernel. The main contribution of Theorem 1 is that it relaxes the PSD assumption of the spectral density, thus the induced NG-SM kernel can mitigate the constraint of identical spectral variance.*

Remark 2. *To derive a closed-form feature map for any kernel, one potential approach is to decompose the spectral density* $S(\mathbf{w}_1, \mathbf{w}_2)$ *into some density functions* $p(\mathbf{w}_1, \mathbf{w}_2)$ *[Ton et al., 2018, Samo and Roberts, 2015], such as:*

$$S(\mathbf{w}_1, \mathbf{w}_2) = \frac{1}{4}(p(\mathbf{w}_1, \mathbf{w}_2) + p(\mathbf{w}_2, \mathbf{w}_1) + p(\mathbf{w}_1)\delta(\mathbf{w}_2 - \mathbf{w}_1) + p(\mathbf{w}_2)\delta(\mathbf{w}_1 - \mathbf{w}_2)), \tag{23}$$

where $p(\mathbf{w}_1)$ *and* $p(\mathbf{w}_2)$ *are the marginal distributions of* $p(\mathbf{w}_1, \mathbf{w}_2)$*, and* $\delta(x)$ *denotes the Dirac delta function. Subsequently, MC integration can be applied to* $S(\mathbf{w}_1, \mathbf{w}_2)$ *to derive the closed-form feature map, see details in Appendix C.1.*

B.2 PROOF OF THEOREM 1

($\Longrightarrow$) Suppose there exists a continuous kernel $k(\mathbf{x}_1, \mathbf{x}_2)$ on $\mathbb{R}^D$. By the Theorem 5, this kernel can be represented as:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \int \exp\left(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\right)\, m(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2),$$

where m is the Lebesgue-Stieltjes measure associated with some PSD function $S(\mathbf{w}_1, \mathbf{w}_2)$ of bounded variation.

To ensure that the kernel function is exchangeable and PSD, we design the spectral density $S(\mathbf{w}_1, \mathbf{w}_2)$ as follows:

$$S(\mathbf{w}_1, \mathbf{w}_2) = \frac{1}{4}(p(\mathbf{w}_1, \mathbf{w}_2) + p(\mathbf{w}_2, \mathbf{w}_1) + p(\mathbf{w}_1)\delta(\mathbf{w}_2 - \mathbf{w}_1) + p(\mathbf{w}_2)\delta(\mathbf{w}_1 - \mathbf{w}_2)), \tag{24}$$

where $\delta(x)$ represents the Dirac delta function, and p is a certain density function that can be decomposed from S as Eq. (24). Additionally, $p(\mathbf{w}_1)$ and $p(\mathbf{w}_2)$ are the marginal distributions of $p(\mathbf{w}_1, \mathbf{w}_2)$.

The resulting kernel is

$$\begin{aligned} k\left(\mathbf{x}_1, \mathbf{x}_2\right) =& \frac{1}{4}\Bigg(\int \exp\left(i\left(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2\right)\right) p\left(\mathbf{w}_1, \mathbf{w}_2\right)\, \mathrm{d}\mathbf{w}_1\, \mathrm{d}\mathbf{w}_2 + \int \exp\left(i\left(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2\right)\right) p\left(\mathbf{w}_2, \mathbf{w}_1\right)\, \mathrm{d}\mathbf{w}_2\, \mathrm{d}\mathbf{w}_1 \\ &+ \int \exp\left(i\mathbf{w}_1^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) p\left(\mathbf{w}_1\right) \delta(\mathbf{w}_2 - \mathbf{w}_1)\, \mathrm{d}\mathbf{w}_1 + \int \exp\left(i\mathbf{w}_2^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) p\left(\mathbf{w}_2\right) \delta(\mathbf{w}_1 - \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_2\Bigg) \\ =& \frac{1}{4}\Bigg(\int \exp\left(i\left(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2\right)\right) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2) + \int \exp\left(i\left(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2\right)\right) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2) \\ &+ \int \exp\left(i\left(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2\right)\right) u(\mathrm{d}\mathbf{w}_1) + \int \exp\left(i\left(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2\right)\right) u(\mathrm{d}\mathbf{w}_2)\Bigg), \end{aligned} \tag{25}$$

where u is the Lebesgue-Stieltjes measure associated with the density function $p(\mathbf{w}_1, \mathbf{w}_2)$.

Note that we can disregard the δ functions in the last two terms of the expression, as the integrands in these terms depend solely on $\mathbf{w}_1$ or $\mathbf{w}_2$. Consequently, we can only integrate over the single variable, while setting the other variable to be equal to the one being integrated.

Finally, we can express the kernel $k\left(\mathbf{x}_1, \mathbf{x}_2\right) =$

$$\frac{1}{4}\int (\exp\left(i\left(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2\right)\right) + \exp\left(i\left(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2\right)\right) + \exp\left(i\mathbf{w}_1^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) + \exp\left(i\mathbf{w}_2^\top(\mathbf{x}_1 - \mathbf{x}_2)\right)) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2), \tag{26}$$

which is the expression shown in Theorem 1.

($\Longleftarrow$) Given the function $k\left(\mathbf{x}_1, \mathbf{x}_2\right) =$

$$\frac{1}{4}\int (\exp\left(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\right) + \exp\left(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\right) + \exp\left(i\mathbf{w}_1^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) + \exp\left(i\mathbf{w}_2^\top(\mathbf{x}_1 - \mathbf{x}_2)\right)) u(\mathrm{d}\mathbf{w}_1, \mathrm{d}\mathbf{w}_2), \tag{27}$$

we have the condition that:

$$S(\mathbf{w}_1, \mathbf{w}_2) = \frac{1}{4}(p(\mathbf{w}_1, \mathbf{w}_2) + p(\mathbf{w}_2, \mathbf{w}_1) + p(\mathbf{w}_1)\delta(\mathbf{w}_2 - \mathbf{w}_1) + p(\mathbf{w}_2)\delta(\mathbf{w}_1 - \mathbf{w}_2)), \tag{28}$$

is a PSD function. We aim to demonstrate that $k(\mathbf{x}_1, \mathbf{x}_2)$ is a valid kernel function. Specifically, we need to show that k is both symmetric and PSD.

First, it is straightforward to show that $k(\mathbf{x}_1, \mathbf{x}_2) = k(\mathbf{x}_2, \mathbf{x}_1)$, confirming that k is symmetric.

The next step is to establish that k is PSD. Consider that $k(\mathbf{x}_1, \mathbf{x}_2) =$

$$\begin{aligned}
&\frac{1}{4}\Big(\int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_1, \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 + \int \exp\big(i(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2)\big) p(\mathbf{w}_1, \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 \\
&\quad + \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_1^\top \mathbf{x}_2)\big) p(\mathbf{w}_1, \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 + \int \exp\big(i(\mathbf{w}_2^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_1, \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2\Big) \\
&= \frac{1}{4}\Big(\int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_1, \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 + \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_2, \mathbf{w}_1)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 \\
&\quad + \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_1)\delta(\mathbf{w}_2 - \mathbf{w}_1)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 + \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) p(\mathbf{w}_2)\delta(\mathbf{w}_1 - \mathbf{w}_2)\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2\Big) \\
&= \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) \frac{1}{4}(p(\mathbf{w}_1, \mathbf{w}_2) + p(\mathbf{w}_2, \mathbf{w}_1) + p(\mathbf{w}_1)\delta(\mathbf{w}_2 - \mathbf{w}_1) + p(\mathbf{w}_2)\delta(\mathbf{w}_1 - \mathbf{w}_2))\, \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 \\
&= \int \exp\big(i(\mathbf{w}_1^\top \mathbf{x}_1 - \mathbf{w}_2^\top \mathbf{x}_2)\big) m(\mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2),
\end{aligned} \tag{29}$$

where m is the Lebesgue-Stieltjes measure associated with the PSD density function $S(\mathbf{w}_1, \mathbf{w}_2)$. Thus, by Theorem 5, $k(\mathbf{w}_1, \mathbf{w}_2)$ is PSD. □

B.3 DERIVATION OF NEXT-GEN SM KERNEL

The BSM kernel based on Theorem 5 is constrained by the requirement of identical variances for $\mathbf{w}_1$ and $\mathbf{w}_2$, and is incompatible with the RFF approximation technique. In this section, we derive the NG-SM kernel based on Theorem 1, which effectively resolves these limitations.

The spectral density of the NG-SM kernel is designed as:

$$p_{\text{ngsm}}(\mathbf{w}_1, \mathbf{w}_2) = \sum_{q=1}^{Q} \alpha_q s_q(\mathbf{w}_1, \mathbf{w}_2), \tag{30}$$

with each

$$s_q(\mathbf{w}_1, \mathbf{w}_2) = \frac{1}{2}\mathcal{N}\left(\begin{pmatrix}\mathbf{w}_1 \\ \mathbf{w}_2\end{pmatrix} \middle| \begin{pmatrix}\boldsymbol{\mu}_{q1} \\ \boldsymbol{\mu}_{q2}\end{pmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\mathrm{c}}^\top \\ \boldsymbol{\Sigma}_{\mathrm{c}} & \boldsymbol{\Sigma}_2\end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\begin{pmatrix}-\mathbf{w}_1 \\ -\mathbf{w}_2\end{pmatrix} \middle| \begin{pmatrix}\boldsymbol{\mu}_{q1} \\ \boldsymbol{\mu}_{q2}\end{pmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\mathrm{c}}^\top \\ \boldsymbol{\Sigma}_{\mathrm{c}} & \boldsymbol{\Sigma}_2\end{bmatrix}\right). \tag{31}$$

We simplify the notation by omitting the index q from the sub-matrices $\boldsymbol{\Sigma}_1 = \mathrm{diag}(\boldsymbol{\sigma}_{q1}^2)$, $\boldsymbol{\Sigma}_2 = \mathrm{diag}(\boldsymbol{\sigma}_{q2}^2)$, and $\boldsymbol{\Sigma}_{\mathrm{c}} = \rho_q \,\mathrm{diag}(\boldsymbol{\sigma}_{q1})\,\mathrm{diag}(\boldsymbol{\sigma}_{q2})$, where $\boldsymbol{\sigma}_{q1}^2, \boldsymbol{\sigma}_{q2}^2 \in \mathbb{R}^D$ and ρ_q represents the correlation between $\mathbf{w}_1$ and $\mathbf{w}_2$. These terms together define the covariance matrix for the q-th bivariate Gaussian component. Additionally, the vectors $\boldsymbol{\mu}_{q1}$ and $\boldsymbol{\mu}_{q2} \in \mathbb{R}^D$ serve as the mean of the q-th bivariate Gaussian component.

Relying on Theorem 1, we derive the NG-SM kernel as following:

$$
\begin{aligned}
k_{\text{ngsm}}\left(\mathbf{x}_1, \mathbf{x}_2\right) =& \frac{1}{4} \int p_{\text{ngsm}}\left(\mathbf{w}_1, \mathbf{w}_2\right) \Big(\exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) \\
& + \exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big)\Big) \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 \\
=& \frac{1}{4} \sum_{q=1}^{Q} \alpha_q \int \frac{1}{2}\left(\Phi_q(\mathbf{w}_1, \mathbf{w}_2) + \Phi_q(-\mathbf{w}_1, -\mathbf{w}_2)\right) \Big(\exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big) \\
& + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big)\Big) \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2,
\end{aligned}
$$

where

$$
\Phi_q(\mathbf{w}_1, \mathbf{w}_2) = \mathcal{N}\left(\begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{pmatrix} \middle| \begin{pmatrix} \boldsymbol{\mu}_{q1} \\ \boldsymbol{\mu}_{q2} \end{pmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\mathrm{c}}^\top \\ \boldsymbol{\Sigma}_{\mathrm{c}} & \boldsymbol{\Sigma}_2 \end{bmatrix} \right). \tag{32}
$$

We focus solely on the real part of the kernel function. Since the real part of the integrand is a cosine function, and both Φ_q and the cosine function are even functions, we can therefore simplify the expression as follows.

$$
\begin{aligned}
=& \frac{1}{4} \sum_{q=1}^{Q} \alpha_q \int \Phi_q(\mathbf{w}_1, \mathbf{w}_2) \Big(\exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) \\
& + \exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) + \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big)\Big) \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2 \\
=& \frac{1}{4} \sum_{q=1}^{Q} \alpha_q \Big(\underbrace{\int \Phi_q(\mathbf{w}_1, \mathbf{w}_2) \exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big) \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2}_{\text{Term (1)}} + \underbrace{\int \Phi_q(\mathbf{w}_2, \mathbf{w}_1) \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) \mathrm{d}\mathbf{w}_1 \mathrm{d}\mathbf{w}_2}_{\text{Term (2)}} \\
& + \underbrace{\int \Phi_q(\mathbf{w}_1) \exp\big(i\mathbf{w}_1^\top \mathbf{x}_1 - i\mathbf{w}_1^\top \mathbf{x}_2\big) \mathrm{d}\mathbf{w}_1}_{\text{Term (3)}} + \underbrace{\int \Phi_q(\mathbf{w}_2) \exp\big(i\mathbf{w}_2^\top \mathbf{x}_1 - i\mathbf{w}_2^\top \mathbf{x}_2\big) \mathrm{d}\mathbf{w}_2}_{\text{Term (4)}} \Big),
\end{aligned} \tag{33}
$$

where $\Phi_q(\mathbf{w}_1)$ and $\Phi_q(\mathbf{w}_2)$ are marginal distributions of $\Phi_q(\mathbf{w}_1, \mathbf{w}_2)$. Next, we will derive the closed forms for each term. First, Term (1) is given by:

$$
\begin{aligned}
\text{Term (1)} =& \int \mathcal{N}\left(\mathbf{w} \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right) e^{\mathbf{w}^\top \tilde{\mathbf{x}}} \mathrm{d}\mathbf{w} \\
=& \frac{1}{(2\pi)^2 |\boldsymbol{\Sigma}_q|} \int \exp\left(-\frac{1}{2} \left(\mathbf{w} - \boldsymbol{\mu}_q\right)^\top \boldsymbol{\Sigma}_q^{-1} \left(\mathbf{w} - \boldsymbol{\mu}_q\right) + \mathbf{w}^\top \tilde{\mathbf{x}} \right) \mathrm{d}\mathbf{w} \\
=& \frac{1}{(2\pi)^2 |\boldsymbol{\Sigma}_q|} \int \exp\left(-\frac{1}{2} \mathbf{w}^\top \boldsymbol{\Sigma}_q^{-1} \mathbf{w} + \mathbf{w}^\top \left(\tilde{\mathbf{x}} + \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q\right) - \frac{1}{2} \boldsymbol{\mu}_q^\top \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q \right) \mathrm{d}\mathbf{w} \\
=& \exp\left(\frac{1}{2} \left(\tilde{\mathbf{x}} + \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q^\top\right)^\top \boldsymbol{\Sigma}_q \left(\tilde{\mathbf{x}} + \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q^\top\right) \right) \exp\left(-\frac{1}{2} \boldsymbol{\mu}_q^\top \boldsymbol{\Sigma}_q^{-1} \boldsymbol{\mu}_q \right) \\
=& \exp\left(\frac{1}{2} \tilde{\mathbf{x}}^\top \boldsymbol{\Sigma}_q \tilde{\mathbf{x}} + \boldsymbol{\mu}_q^\top \tilde{\mathbf{x}} \right),
\end{aligned} \tag{34}
$$

where we defined $\tilde{\mathbf{x}} = (i\mathbf{x}_1, -i\mathbf{x}_2)$ and $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$. In addition, $\boldsymbol{\mu}_q = (\boldsymbol{\mu}_{q1}, \boldsymbol{\mu}_{q2})$ and

$$
\boldsymbol{\Sigma}_q = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{\mathrm{c}}^\top \\ \boldsymbol{\Sigma}_{\mathrm{c}} & \boldsymbol{\Sigma}_2 \end{bmatrix}.
$$

The first term of the kernel mixture is then given by:

$$
\begin{aligned}
\exp\left(\frac{1}{2} \tilde{\mathbf{x}}^\top \boldsymbol{\Sigma}_q \tilde{\mathbf{x}} + \boldsymbol{\mu}_q^\top \tilde{\mathbf{x}} \right) =& \exp\left(-\frac{1}{2} \left(\mathbf{x}_1^\top \boldsymbol{\Sigma}_1 \mathbf{x}_1 - 2\mathbf{x}_1^\top \boldsymbol{\Sigma}_c \mathbf{x}_2 + \mathbf{x}_2^\top \boldsymbol{\Sigma}_2 \mathbf{x}_2 \right) \right) \exp\left(i \left(\boldsymbol{\mu}_{q1}^\top \mathbf{x}_1 - \boldsymbol{\mu}_{q2}^\top \mathbf{x}_2 \right) \right) \\
=& \exp\left(-\frac{1}{2} \left(\mathbf{x}_1^\top \boldsymbol{\Sigma}_1 \mathbf{x}_1 - 2\mathbf{x}_1^\top \boldsymbol{\Sigma}_c \mathbf{x}_2 + \mathbf{x}_2^\top \boldsymbol{\Sigma}_2 \mathbf{x}_2 \right) \right) \cos\left(\boldsymbol{\mu}_{q1}^\top \mathbf{x}_1 - \boldsymbol{\mu}_{q2}^\top \mathbf{x}_2 \right).
\end{aligned} \tag{35}
$$

By swapping $\mathbf{x}_1$ and $\mathbf{x}_2$ in Term (1), the closed form of Term (2) can be easily obtained as below:

$$\text{Term (2)} = \exp\left(-\frac{1}{2}\left(\mathbf{x}_2\boldsymbol{\Sigma}_1\mathbf{x}_2^\top - 2\mathbf{x}_1\boldsymbol{\Sigma}_c\mathbf{x}_2^\top + \mathbf{x}_1\boldsymbol{\Sigma}_2\mathbf{x}_1^\top\right)\right)\cos\left(\boldsymbol{\mu}_{q1}\mathbf{x}_2^\top - \boldsymbol{\mu}_{q2}\mathbf{x}_1^\top\right). \tag{36}$$

Term (3) of the kernel is then given by:

$$\begin{aligned}
\text{Term (3)} =& \int \mathcal{N}\left(\mathbf{w}_1 \mid \boldsymbol{\mu}_{q1}, \boldsymbol{\Sigma}_1\right)\exp\big(i\mathbf{w}_1^\top(\mathbf{x}_1 - \mathbf{x}_2)\big)\mathrm{d}\mathbf{w}_1 \\
=& \frac{1}{(2\pi)^2|\boldsymbol{\Sigma}_1|}\int \exp\left(-\frac{1}{2}\left(\mathbf{w}_1 - \boldsymbol{\mu}_{q1}\right)^\top\boldsymbol{\Sigma}_1^{-1}\left(\mathbf{w}_1 - \boldsymbol{\mu}_{q1}\right) + i\mathbf{w}_1^\top(\mathbf{x}_1 - \mathbf{x}_2)\right)\mathrm{d}\mathbf{w}_1 \\
=& \frac{1}{(2\pi)^2|\boldsymbol{\Sigma}_1|}\int \exp\left(-\frac{1}{2}\mathbf{w}_1^\top\boldsymbol{\Sigma}_1^{-1}\mathbf{w}_1 + \mathbf{w}_1^\top\left(i(\mathbf{x}_1 - \mathbf{x}_2) + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_{q1}\right) - \frac{1}{2}\boldsymbol{\mu}_{q1}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_{q1}\right)\mathrm{d}\mathbf{w}_1 \\
=& \exp\left(\frac{1}{2}\left(i(\mathbf{x}_1 - \mathbf{x}_2) + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_{q1}\right)^\top\boldsymbol{\Sigma}_1\left(i(\mathbf{x}_1 - \mathbf{x}_2) + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_{q1}\right)\right)\exp\left(-\frac{1}{2}\boldsymbol{\mu}_{q1}^\top\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_{q1}\right) \\
=& \exp\left(\frac{1}{2}i(\mathbf{x}_1 - \mathbf{x}_2)^\top\boldsymbol{\Sigma}_1 i(\mathbf{x}_1 - \mathbf{x}_2) + \boldsymbol{\mu}_{q1}^\top i(\mathbf{x}_1 - \mathbf{x}_2)\right) \\
=& \exp\left(-\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^\top\boldsymbol{\Sigma}_1(\mathbf{x}_1 - \mathbf{x}_2)\right)\right)\exp\left(i\boldsymbol{\mu}_{q1}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) \\
=& \exp\left(-\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^\top\boldsymbol{\Sigma}_1(\mathbf{x}_1 - \mathbf{x}_2)\right)\right)\cos\left(\boldsymbol{\mu}_{q1}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right).
\end{aligned} \tag{37}$$

The 4'th term of the kernel can be derived in a manner similar to the 3'rd term.

$$\text{Term (4)} = \exp\left(-\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^\top\boldsymbol{\Sigma}_2(\mathbf{x}_1 - \mathbf{x}_2)\right)\right)\cos\left(\boldsymbol{\mu}_{q2}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right). \tag{38}$$

Thus, NG-SM kernel takes the form:

$$\begin{aligned}
k(\mathbf{x}_1, \mathbf{x}_2) =& \frac{1}{4}\sum_{q=1}^{Q}\alpha_q\left[\exp\left(-\frac{1}{2}\left(\mathbf{x}_1^\top\boldsymbol{\Sigma}_1\mathbf{x}_1 - 2\mathbf{x}_1^\top\boldsymbol{\Sigma}_c\mathbf{x}_2 + \mathbf{x}_2^\top\boldsymbol{\Sigma}_2\mathbf{x}_2\right)\right)\cos\left(\boldsymbol{\mu}_{q1}^\top\mathbf{x}_1 - \boldsymbol{\mu}_{q2}^\top\mathbf{x}_2\right)\right. \\
&+ \exp\left(-\frac{1}{2}\left(\mathbf{x}_2^\top\boldsymbol{\Sigma}_1\mathbf{x}_2 - 2\mathbf{x}_1^\top\boldsymbol{\Sigma}_c\mathbf{x}_2 + \mathbf{x}_1^\top\boldsymbol{\Sigma}_2\mathbf{x}_1\right)\right)\cos\left(\boldsymbol{\mu}_{q1}^\top\mathbf{x}_2 - \boldsymbol{\mu}_{q2}^\top\mathbf{x}_1\right) \\
&+ \exp\left(-\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^\top\boldsymbol{\Sigma}_1(\mathbf{x}_1 - \mathbf{x}_2)\right)\right)\cos\left(\boldsymbol{\mu}_{q1}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right) \\
&\left.+ \exp\left(-\frac{1}{2}\left(\mathbf{x}_1 - \mathbf{x}_2\right)^\top\boldsymbol{\Sigma}_2(\mathbf{x}_1 - \mathbf{x}_2)\right)\right)\cos\left(\boldsymbol{\mu}_{q2}^\top(\mathbf{x}_1 - \mathbf{x}_2)\right)\right].
\end{aligned} \tag{39}$$

C AUTO-DIFFERENTIABLE NEXT-GEN SM KERNEL USING RFF APPROXIMATION

C.1 RANDOM FOURIER FEATURE FOR NEXT-GEN SM KERNEL

Our proposed Theorem 1 establishes the following duality: $k\left(\mathbf{x}_1, \mathbf{x}_2\right)$

$$= \frac{1}{4}\mathbb{E}_u\left(\exp\big(i(\mathbf{w}_1^\top\mathbf{x}_1 - \mathbf{w}_2^\top\mathbf{x}_2)\big) + \exp\big(i(\mathbf{w}_2^\top\mathbf{x}_1 - \mathbf{w}_1^\top\mathbf{x}_2)\big) + \exp\big(i(\mathbf{w}_1^\top\mathbf{x}_1 - \mathbf{w}_1^\top\mathbf{x}_2)\big) + \exp\big(i(\mathbf{w}_2^\top\mathbf{x}_1 - \mathbf{w}_2^\top\mathbf{x}_2)\big)\right).$$

By estimating this expectation with the MC estimator using spectral points $\{\mathbf{w}_1^{(l)}; \mathbf{w}_2^{(l)}\}_{l=1}^{L/2}$ sampled from $p(\mathbf{w}_1, \mathbf{w}_2)$, we can drive

$$
\begin{aligned}
k\left(\mathbf{x}_1, \mathbf{x}_2\right) \approx & \frac{1}{2L} \sum_{l=1}^{L/2} \Big(\exp\Big(i(\mathbf{w}_1^{(l)\top} \mathbf{x}_1 - \mathbf{w}_2^{(l)\top} \mathbf{x}_2) \Big) + \exp\Big(i(\mathbf{w}_2^{(l)\top} \mathbf{x}_1 - \mathbf{w}_1^{(l)\top} \mathbf{x}_2) \Big) \\
& + \exp\Big(i(\mathbf{w}_1^{(l)\top} \mathbf{x}_1 - \mathbf{w}_1^{(l)\top} \mathbf{x}_2) \Big) + \exp\Big(i(\mathbf{w}_2^{(l)\top} \mathbf{x}_1 - \mathbf{w}_2^{(l)\top} \mathbf{x}_2) \Big) \Big) \\
= & \frac{1}{2L} \sum_{l=1}^{L/2} \Big(\cos\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\Big) \cos\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_2\Big) + \cos\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\Big) \cos\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\Big) \\
& + \cos\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_1\Big) \cos\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_2\Big) + \cos\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_1\Big) \cos\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\Big) \\
& + \sin\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\Big) \sin\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_2\Big) + \sin\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_1\Big) \sin\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\Big) \\
& + \sin\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_1\Big) \sin\Big(\mathbf{w}_1^{(l)\top} \mathbf{x}_2\Big) + \sin\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_1\Big) \sin\Big(\mathbf{w}_2^{(l)\top} \mathbf{x}_2\Big) \Big) \\
= & \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2)
\end{aligned}
\tag{40}
$$

where

$$
\phi\left(\mathbf{x}\right) = \sqrt{\frac{1}{2L}} \begin{bmatrix} \cos\left(\mathbf{w}_1^{(1:L/2)\top} \mathbf{x}\right) + \cos\left(\mathbf{w}_2^{(1:L/2)\top} \mathbf{x}\right) \\ \sin\left(\mathbf{w}_1^{(1:L/2)\top} \mathbf{x}\right) + \sin\left(\mathbf{w}_2^{(1:L/2)\top} \mathbf{x}\right) \end{bmatrix} \in \mathbb{R}^L. \tag{41}
$$

Here the superscript $1 : L/2$ indicates that the cosine plus cosine or sine plus sine function is repeated $L/2$ times, with each element corresponding to the one entry of $\{\mathbf{w}_1^{(l)}; \mathbf{w}_2^{(l)}\}_{l=1}^{L/2}$. If we specify the spectral density as $p_{\text{ngsm}}(\mathbf{w}_1, \mathbf{w}_2)$ (Eq. (30)), the estimator of the kernel $k_{\text{ngsm}}(\mathbf{x}_1, \mathbf{x}_2)$ can be formulated as:

$$
k_{\text{ngsm}}(\mathbf{x}_1, \mathbf{x}_2) \approx \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2), \tag{42}
$$

where random features $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$ are constructed using spectral points sampled from $p_{\text{ngsm}}(\mathbf{w}_1, \mathbf{w}_2)$.

C.2 ELBO DERIVATION AND EVALUATION

The term (a) of ELBO is handled numerically with MC estimation as below:

$$
\begin{aligned}
\text{(a)} &= \sum_{m=1}^{M_v} \mathbb{E}_{q(\mathbf{X}, \mathbf{W})} \left[\log p(\mathbf{y}^v_{:,m} | \mathbf{X}, \mathbf{W}^v) \right] && \text{(43a)} \\
&\approx \sum_{m=1}^{M_v} \frac{1}{I} \sum_{i=1}^{I} \log \mathcal{N}(\mathbf{y}^v_{:,m} | \mathbf{0}, \tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}} + \sigma_v^2 \mathbf{I}_N), && \text{(43b)}
\end{aligned}
$$

where I denotes the number of MC samples drawn from $q(\mathbf{X}, \mathbf{W})$. Additionally, $\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}} = (\mathbf{\Phi}^v_x \mathbf{\Phi}^{v\top}_x)^{(i)}$ is the NG-SM kernel gram matrix approximation, where $\Phi^v_x \in \mathbb{R}^{N \times L}$.

The term (b) of ELBO can be evaluated analytically due to the Gaussian nature of the distributions. More specific, we have

$$
\begin{aligned}
\text{(b)} &= \sum_{n=1}^{N} \text{KL}(q(\mathbf{x}_n) \| p(\mathbf{x}_n)) && \text{(44a)} \\
&= \frac{1}{2} \sum_{n=1}^{N} \Big[\text{tr}(\mathbf{S}_n) + \boldsymbol{\mu}_n^\top \boldsymbol{\mu}_n - \log |\mathbf{S}_n| - D \Big], && \text{(44b)}
\end{aligned}
$$

where D represents the dimensionality of $\mathbf{x}_n$, and $\mathbf{S}_n$ is commonly assumed to be a diagonal matrix. Consequently, the ELBO can be expressed as follows:

$$
\begin{aligned}
ELBO &= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\frac{p(\mathbf{Y},\mathbf{X};\mathbf{W})}{q(\mathbf{X},\mathbf{W})}\right] \\
&= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\log\frac{p(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^v)p(\mathbf{Y}^v|\mathbf{X},\mathbf{W}^v)}{q(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^v)}\right] \\
&= \sum_{v=1}^{V}\underbrace{\mathbb{E}_{q(\cdot,\cdot)}\left[\log p(\mathbf{Y}^v|\mathbf{X},\mathbf{W}^v)\right]}_{\text{(a): reconstruction}} - \underbrace{\mathrm{KL}(q(\mathbf{X})\|p(\mathbf{X}))}_{\text{(b): regularization}} \\
&\approx \sum_{v=1}^{V}\sum_{m=1}^{M_v}\frac{1}{I}\sum_{i=1}^{I}\log\mathcal{N}(\mathbf{y}^v_{:,m}|\mathbf{0},\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma^2\mathbf{I}_N) - \frac{1}{2}\sum_{n=1}^{N}\left[\mathrm{tr}(\mathbf{S}_n)+\boldsymbol{\mu}_n^\top\boldsymbol{\mu}_n-\log|\mathbf{S}_n|-D\right] \\
&= \sum_{v=1}^{V}\sum_{m=1}^{M_v}\frac{1}{I}\sum_{i=1}^{I}\left\{-\frac{N}{2}\log 2\pi-\frac{1}{2}\log\left|\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma_v^2\mathbf{I}_N\right|-\frac{1}{2}\mathbf{y}^{v\top}_{:,m}\left(\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma_v^2\mathbf{I}_N\right)^{-1}\mathbf{y}^v_{:,m}\right\} \\
&\quad -\frac{1}{2}\sum_{n=1}^{N}\left[\mathrm{tr}(\mathbf{S}_n)+\boldsymbol{\mu}_n^\top\boldsymbol{\mu}_n-\log|\mathbf{S}_n|-D\right].
\end{aligned}
$$

When $N \gg L$, both the determinant and the inverse of $\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma_v^2\mathbf{I}_N$ can be computed efficiently by the following two lemma [Williams and Rasmussen, 2006].

Lemma 1. *Suppose* $\mathbf{A}$ *is an invertible* n*-by-*n *matrix and* $\mathbf{U},\mathbf{V}$ *are* n*-by-*m *matrices. Then the following determinant equality holds.*

$$
\left|\mathbf{A}+\mathbf{U}\mathbf{V}^\top\right| = \left|\mathbf{I}_{\mathrm{m}}+\mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U}\right|\left|\mathbf{A}\right|.
$$

Lemma 2 (Woodbury matrix identity)**.** *Suppose* $\mathbf{A}$ *is an invertible* n*-by-*n *matrix and* $\mathbf{U},\mathbf{V}$ *are* n*-by-*m *matrices. Then*

$$
\left(\mathbf{A}+\mathbf{U}\mathbf{V}^\top\right)^{-1} = \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{U}(\mathbf{I}_{\mathrm{m}}+\mathbf{V}^\top\mathbf{U})^{-1}\mathbf{V}^\top.
$$

We can reduce the computational complexity for evaluating the ELBO from the original $\mathcal{O}(N^3)$ to $\mathcal{O}(NL^2)$ as below:

$$
\left|\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma_v^2\mathbf{I}_N\right| = \sigma_v^{2N}\left|\mathbf{I}_L+\frac{1}{\sigma_v^2}(\mathbf{\Phi}_x^v\mathbf{\Phi}_x^{v\top})^{(i)}\right|, \tag{45}
$$

$$
\left(\tilde{\mathbf{K}}^{v(i)}_{\text{ngsm}}+\sigma_v^2\mathbf{I}_N\right)^{-1} = \frac{1}{\sigma_v^2}\left[\mathbf{I}_N-\mathbf{\Phi}_x^{v(i)}\left(\mathbf{I}_L+(\mathbf{\Phi}_x^v\mathbf{\Phi}_x^{v\top})^{(i)}\right)^{-1}\mathbf{\Phi}_x^{v(i)\top}\right]. \tag{46}
$$

C.3 TREATING W VARIATIONALLY

Alternatively, we define the variational distributions as

$$
q(\mathbf{X},\mathbf{W}) = q(\mathbf{W};\boldsymbol{\eta})q(\mathbf{X}),
$$

where the variational distribution $q(\mathbf{W};\boldsymbol{\eta})$ is also a bivariate Gaussian mixture that is parameterized by $\boldsymbol{\eta}$. By combining these variational distributions with the joint distribution defined in Eq. (11), we derive the following ELBO:

$$
\begin{aligned}
ELBO &= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\frac{p(\mathbf{Y},\mathbf{X},\mathbf{W})}{q(\mathbf{W};\boldsymbol{\eta})q(\mathbf{X})}\right] \\
&= \mathbb{E}_{q(\mathbf{X},\mathbf{W})}\left[\log\frac{p(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^v;\boldsymbol{\theta}_{\text{ngsm}})p(\mathbf{Y}^v|\mathbf{X},\mathbf{W}^v)}{q(\mathbf{X})\prod_{v=1}^{V}p(\mathbf{W}^v;\boldsymbol{\eta})}\right] \\
&= \sum_{v=1}^{V}\underbrace{\mathbb{E}_{q(\cdot,\cdot)}\left[\log p(\mathbf{Y}^v|\mathbf{X},\mathbf{W}^v)\right]}_{\text{(a): reconstruction}} - \underbrace{\mathrm{KL}(q(\mathbf{X})\|p(\mathbf{X}))}_{\text{(b): regularization of }\mathbf{X}} - \underbrace{\mathrm{KL}(q(\mathbf{W};\boldsymbol{\eta})\|p(\mathbf{W};\boldsymbol{\theta}_{\text{ngsm}}))}_{\text{(c): regularization of }\mathbf{W}},
\end{aligned} \tag{47}
$$

where we redefine the prior distribution $p(\mathbf{W})$ as $p(\mathbf{W};\boldsymbol{\theta}_{\text{ngsm}})$ to maintain notational consistency. When maximizing the ELBO, we obtain $q(\mathbf{W};\boldsymbol{\eta}) = p(\mathbf{W};\boldsymbol{\theta}_{\text{ngsm}})$, as the optimization variable $\boldsymbol{\theta}_{\text{ngsm}}$ only affects term c. Consequently, term (c) becomes zero, and Eq. (47) aligns with the optimization objective outlined in the main text.

C.4 PROOF OF PROPOSITION 1

Proof. The proposed two-step reparameterization trick leverages sequential simulation, where $\mathbf{w}_{q1}$ is first sampled from $p(\mathbf{w}_{q1})$, followed by $\mathbf{w}_{q2}$ drawn from the conditional distribution $p(\mathbf{w}_{q2}|\mathbf{w}_{q1})$.

1) Sample from $p(\mathbf{w}_{q1})$: Given that $\mathbf{w}_{q1}$ follows a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{q1}, \mathrm{diag}(\boldsymbol{\sigma}_{q1}^2))$, we can directly use the reparameterization trick to sample from it, i.e.,

$$\mathbf{w}_{q1}^{(l)} = \boldsymbol{\mu}_{q1} + \boldsymbol{\sigma}_{q1} \circ \boldsymbol{\epsilon}_1,$$

where $\boldsymbol{\epsilon}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2) Sample from $p(\mathbf{w}_{q2}|\mathbf{w}_{q1})$: Given $\mathbf{w}_{q1}^{(l)}$, we use the given correlation parameter ρ_q and a new standard normal variable $\boldsymbol{\epsilon}_2$ to generate $\mathbf{w}_{q2}^{(l)}$:

$$\mathbf{w}_{q2}^{(l)} = \boldsymbol{\mu}_{q2} + \rho_q \boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1} \circ (\mathbf{w}_{q1}^{(l)} - \boldsymbol{\mu}_{q1}) + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2.$$

Now, we need to proof that the generated $\mathbf{w}_{q1}^{(l)}$ and $\mathbf{w}_{q2}^{(l)}$ follow the bivariate Gaussian distribution $s_q(\mathbf{w}_1, \mathbf{w}_2)$. To this ends, we compute the mean and variance of $\mathbf{w}_{q2}^{(l)}$, and the covariance between $\mathbf{w}_{q1}^{(l)}$ and $\mathbf{w}_{q2}^{(l)}$ below.

- Mean of $\mathbf{w}_{q2}^{(l)}$:

$$\mathbb{E}[\mathbf{w}_{q2}^{(l)}] = \mathbb{E}\left[\boldsymbol{\mu}_{q2} + \rho_q \boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1} \circ (\mathbf{w}_{q1}^{(l)} - \boldsymbol{\mu}_{q1}) + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2\right].$$

Since $\mathbb{E}[\boldsymbol{\epsilon}_2] = 0$ and $\mathbb{E}[\mathbf{w}_{q1}^{(l)}] = \boldsymbol{\mu}_{q1}$, we have:

$$\mathbb{E}[\mathbf{w}_{q2}^{(l)}] = \boldsymbol{\mu}_{q2} + \rho_q \boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1} \circ (\mathbb{E}[\mathbf{w}_{q1}^{(l)}] - \boldsymbol{\mu}_{q1}) + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \mathbb{E}[\boldsymbol{\epsilon}_2] = \boldsymbol{\mu}_{q2}.$$

- Variance of $\mathbf{w}_{q2}^{(l)}$:

$$\mathrm{Var}(\mathbf{w}_{q2}^{(l)}) = \mathrm{Var}\left(\rho_q \boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1} \circ (\mathbf{w}_{q1}^{(l)} - \boldsymbol{\mu}_{q1}) + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2\right).$$

Since $\mathbf{w}_{q1}^{(l)} - \boldsymbol{\mu}_{q1}$ and $\boldsymbol{\epsilon}_2$ are independent, and Var$(\boldsymbol{\epsilon}_2) = \mathbf{I}$, we have:

$$\mathrm{Var}(\mathbf{w}_{q2}^{(l)}) = \rho_q^2 \left(\boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1}\right)^2 \circ \mathrm{Var}(\mathbf{w}_{q1}^{(l)}) + (1-\rho_q^2)\boldsymbol{\sigma}_{q2}^2 = \rho_q^2 \boldsymbol{\sigma}_{q2}^2 + (1-\rho_q^2)\boldsymbol{\sigma}_{q2}^2 = \boldsymbol{\sigma}_{q2}^2.$$

- Covariance between $\mathbf{w}_{q1}^{(l)}$ and $\mathbf{w}_{q2}^{(l)}$:

$$\mathrm{Cov}(\mathbf{w}_{q1}^{(l)}, \mathbf{w}_{q2}^{(l)}) = \mathrm{Cov}\left(\boldsymbol{\mu}_{q1} + \boldsymbol{\sigma}_{q1} \circ \boldsymbol{\epsilon}_1, \boldsymbol{\mu}_{q2} + \rho_q \boldsymbol{\sigma}_{q2} \backslash \boldsymbol{\sigma}_{q1} \circ (\mathbf{w}_{q1}^{(l)} - \boldsymbol{\mu}_{q1}) + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2\right). \quad (48)$$

Since $\boldsymbol{\mu}_{q1}$ and $\boldsymbol{\mu}_{q2}$ are constants, the covariance only depends on $\boldsymbol{\sigma}_{q1} \circ \boldsymbol{\epsilon}_1$ and $\rho_q \boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_1 + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2$. Thus we can reformulate Eq. (48) as

$$\mathrm{Cov}(\boldsymbol{\sigma}_{q1} \circ \boldsymbol{\epsilon}_1, \rho_q \boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_1 + \sqrt{1-\rho_q^2}\boldsymbol{\sigma}_{q2} \circ \boldsymbol{\epsilon}_2) = \boldsymbol{\sigma}_{q1}\rho_q \circ \boldsymbol{\sigma}_{q2}\mathrm{Cov}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_1) + \boldsymbol{\sigma}_{q1}\sqrt{1-\rho_q^2} \circ \boldsymbol{\sigma}_{q2}\mathrm{Cov}(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2).$$

Given that $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are independent, Cov$(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2) = \mathbf{0}$, and Cov$(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_1) = \mathbf{I}$, this covariance is equal to:

$$\boldsymbol{\sigma}_{q1}\rho_q \circ \boldsymbol{\sigma}_{q2} \circ \mathbf{I} + \boldsymbol{\sigma}_{q1}\sqrt{1-\rho_q^2} \circ \boldsymbol{\sigma}_{q2} \circ \mathbf{0} = \rho_q \boldsymbol{\sigma}_{q1} \circ \boldsymbol{\sigma}_{q2}.$$

Therefore, the generated $\mathbf{w}_{q1}^{(l)}$ and $\mathbf{w}_{q2}^{(l)}$ follow the bivariate Gaussian distribution $s_q(\mathbf{w}_1, \mathbf{w}_2)$. $\square$

The two-step reparameterization trick simplifies the sampling process of the bivariate Gaussian distribution. Specifically, the traditional sampling method [Mohamed et al., 2020] requires Cholesky decomposition of the $2D \times 2D$ covariance matrix, resulting in a computational complexity of $\mathcal{O}(8D^3)$. In contrast, our method achieves a computational complexity of $\mathcal{O}(D)$ for sampling from $p(\mathbf{w}_{q1})$, while sampling from $p(\mathbf{w}_{q2}|\mathbf{w}_{q1})$ also maintains a complexity of $\mathcal{O}(D)$. This results in a total computational cost of $\mathcal{O}(D)$.

C.5 PROOF OF THEOREM 3

Proof. With the RFF feature map defined in Eq. (15), we can express the inner product of the feature maps as follows:

$$\varphi\left(\mathbf{x}_1\right)^\top \varphi\left(\mathbf{x}_2\right) = \sum_{q=1}^{Q} \alpha_q \sum_{l=1}^{L/2} \frac{1}{2L} A_q^{(l)}, \tag{49}$$

where,

$$\begin{aligned}
A_q^{(l)} = \Big(& \cos\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_1\Big)\cos\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_2\Big) + \cos\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_1\Big)\cos\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_2\Big) \\
&+ \cos\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_1\Big)\cos\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_2\Big) + \cos\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_1\Big)\cos\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_2\Big) \\
&+ \sin\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_1\Big)\sin\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_2\Big) + \sin\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_1\Big)\sin\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_2\Big) \\
&+ \sin\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_1\Big)\sin\Big(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_2\Big) + \sin\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_1\Big)\sin\Big(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_2\Big)\Big),
\end{aligned} \tag{50}$$

where $\{\mathbf{w}_1^l, \mathbf{w}_2^l\}_{l=1}^{L/2}$ are independently and identically distributed (i.i.d.) spectral points drawn from the density function $s_q(\mathbf{w}_1, \mathbf{w}_2)$ using the two step reparameterization trick. Taking the expectation with respect to $p_{\text{ngsm}}(\mathbf{w}_1, \mathbf{w}_2) = \prod_{q=1}^{Q}\prod_{l=1}^{L/2} s_q(\mathbf{w}_1, \mathbf{w}_2)$, we obtain:

$$\begin{aligned}
\mathbb{E}_{p_{\text{ngsm}}(\mathbf{w}_1,\mathbf{w}_2)}\Big[\varphi\left(\mathbf{x}_1\right)^\top \varphi\left(\mathbf{x}_2\right)\Big] &= \mathbb{E}_{p_{\text{ngsm}}(\mathbf{w}_1,\mathbf{w}_2)}\left[\sum_{q=1}^{Q}\alpha_q\sum_{l=1}^{L/2}\frac{1}{2L}A_q^{(l)}\right] \\
&= \sum_{q=1}^{Q}\alpha_q \mathbb{E}_{s\left(\mathbf{w_{q1}}^{1:L/2},\mathbf{w_{q2}}^{1:L/2}\right)}\left[\sum_{l=1}^{L/2}\frac{1}{2L}A_q^{(l)}\right], \qquad \text{(linearity of expectation)}
\end{aligned} \tag{51}$$

$$\begin{aligned}
&= \sum_{q=1}^{Q}\alpha_q\frac{1}{4}\mathbb{E}_{s_q(\mathbf{w}_1,\mathbf{w}_2)}\left[A_q\right] && \text{(i.i.d. of } \mathbf{w}_l^{(q)}) \\
&= \sum_{q=1}^{Q}\alpha_q\frac{1}{4}\mathbb{E}_{s_q(\mathbf{w}_1,\mathbf{w}_2)}\Big[\exp\big(i(\mathbf{w}_1\mathbf{x}_1^\top - \mathbf{w}_2\mathbf{x}_2^\top)\big) + \exp\big(i(\mathbf{w}_2\mathbf{x}_1^\top - \mathbf{w}_1\mathbf{x}_2^\top)\big) \\
&\quad + \exp\big(i(\mathbf{w}_1\mathbf{x}_1^\top - \mathbf{w}_1\mathbf{x}_2^\top)\big) + \exp\big(i(\mathbf{w}_2\mathbf{x}_1^\top - \mathbf{w}_2\mathbf{x}_2^\top)\big)\Big] && \text{(Euler's identity)} \\
&= \sum_{q=1}^{Q}\alpha_q k_q(\mathbf{x}_1,\mathbf{x}_2) \\
&= k_{\text{ngsm}}(\mathbf{x}_1,\mathbf{x}_2). && \text{(NG-SM kernel definition)}
\end{aligned} \tag{52}$$

Thus, we conclude that $\varphi\left(\mathbf{x}_1\right)^\top \varphi\left(\mathbf{x}_2\right)$ provides an unbiased estimator for the NG-SM kernel.

□

C.6 PROOF OF THEOREM 4

Proof. We primarily rely on the Matrix Bernstein inequality [Tropp, 2015] to establish Theorem 4, with the proof outline depicted in Figure 4.

Lemma 3 (Matrix Bernstein Inequality)**.** *Consider a finite sequence $\{\mathbf{E}_i\}$ of independent, random, Hermitian matrices with dimension N. Assume that*

$$\mathbb{E}[\mathbf{E}_i] = \mathbf{0} \textit{ and } \|\mathbf{E}_i\|_2 \leq H \textit{ for each index } i,$$

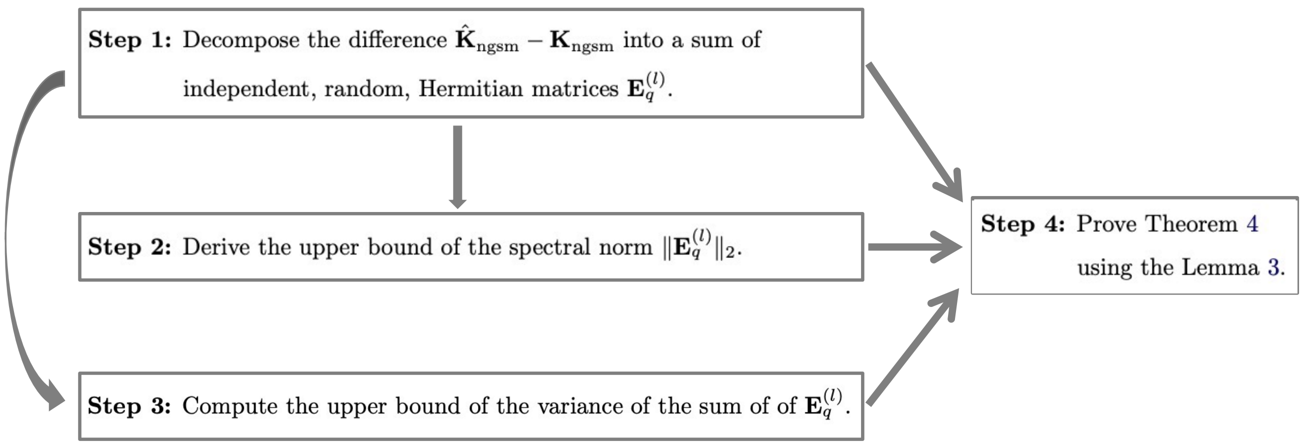


Figure 4: Flowchart for proving Theorem 4

where $\|\cdot\|_2$ denotes the matrix spectral norm. Introduce the random matrix $\mathbf{E} = \sum_i \mathbf{E}_i$, and let $v(\mathbf{E})$ be the matrix variance statistic of the sum:

$$v(\mathbf{E}) = \left\|\mathbb{E}[\mathbf{E}^2]\right\| = \left\|\sum_i \mathbb{E}[\mathbf{E}_i^2]\right\|.$$

Then we have

$$\mathbb{E}\left[\|\mathbf{E}\|_2\right] \leq \sqrt{2v(\mathbf{E}) \log N} + \frac{1}{3} L \log N. \tag{53}$$

Furthermore, for all $\epsilon \geq 0$.

$$P\left\{\|\mathbf{E}\|_2 \geq \epsilon\right\} \leq N \cdot \exp\left(\frac{-\epsilon^2/2}{v(\mathbf{E}) + H\epsilon/3}\right). \tag{54}$$

Proof. The proof of Lemma 3 can be found in Theorem 6.6.1 in Tropp [2015]. □

Step 1: With the constructed NG-SM kernel matrix approximation, $\hat{\mathbf{K}}_{\text{ngsm}} = \Phi_{\text{ngsm}}(\mathbf{X})\Phi_{\text{ngsm}}(\mathbf{X})^\top$, where the random feature matrix $\Phi_{\text{ngsm}}(\mathbf{X}) = \left[\varphi\left(\mathbf{x}_1\right), \ldots, \varphi\left(\mathbf{x}_N\right)\right]^\top \in \mathbb{R}^{N \times QL}$, we have the following approximation error matrix:

$$\mathbf{E} = \hat{\mathbf{K}}_{\text{ngsm}} - \mathbf{K}_{\text{ngsm}}. \tag{55}$$

We are going to show that $\mathbf{E}$ can be factorized as

$$\mathbf{E} = \sum_{q=1}^{Q} \sum_{l=1}^{L/2} \mathbf{E}_q^{(l)}, \tag{56}$$

where $\mathbf{E}_q^{(l)}$ is a sequence of independent, random, Hermitian matrices with dimension N.

Sample $\mathbf{w}_{q1}^{(l)}, \mathbf{w}_{q2}^{(l)}$ from $s_q(\mathbf{w}_1, \mathbf{w}_2)$, and we can show that

$$[\hat{\mathbf{K}}_{\text{ngsm}}]_{h,g} = \sum_{q=1}^{Q} \frac{\alpha_q}{2L} \sum_{l=1}^{L/2} A_q^{(l)}(h, g),$$

where

$$\begin{aligned} A_q^{(l)}(h, g) = \Big(&\cos\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_h\right)\cos\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_g\right) + \cos\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_h\right)\cos\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_g\right) + \cos\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_h\right)\cos\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_g\right) \\ &+ \cos\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_h\right)\cos\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_g\right) + \sin\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_h\right)\sin\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_g\right) + \sin\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_h\right)\sin\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_g\right) \\ &+ \sin\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_h\right)\sin\left(\mathbf{w}_{q1}^{(l)\top}\mathbf{x}_g\right) + \sin\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_h\right)\sin\left(\mathbf{w}_{q2}^{(l)\top}\mathbf{x}_g\right)\Big). \end{aligned} \tag{57}$$

Thus, we have $\hat{\mathbf{K}}_{\text{ngsm}} = \sum_{q=1}^{Q} \sum_{l=1}^{L/2} \frac{\alpha_q}{2L} A_q^{(l)}$. Based on this factorization and Eq. (51) in Proposition 3, we have that

$$\mathbf{K}_{\text{ngsm}} = \sum_{q=1}^{Q} \sum_{l=1}^{L/2} \frac{\alpha_q}{2L} \mathbb{E}[A_q^{(l)}].$$

Therefore, the approximation error matrix $\mathbf{E}$ can be factorized as $\mathbf{E} = \sum_{q=1}^{Q} \sum_{l=1}^{L/2} \mathbf{E}_q^{(l)}$ where

$$\mathbf{E}_q^{(l)} = \frac{\alpha_q}{2L} \left(A_q^{(l)} - \mathbb{E}[A_q^{(l)}] \right) \tag{58}$$

is a sequence of independent, random, Hermitian matrices with dimension N that satisfy the condition of $\mathbb{E}[\mathbf{E}_q^{(l)}] = \mathbf{0}$.

Step 2: We can bound the $\|\mathbf{E}_q^{(l)}\|_2$ by following.

$$\begin{aligned}
\|\mathbf{E}_q^{(l)}\|_2 &= \frac{\alpha_q}{2L} \left\| A_q^{(l)} - \mathbb{E}\left[A_q^{(l)}\right] \right\|_2 && \text{(59a)} \\
&\leq \frac{\alpha_q}{2L} \left(\left\| A_q^{(l)} \right\|_2 + \left\| \mathbb{E}\left[A_q^{(l)}\right] \right\|_2 \right) \qquad \text{(triangle inequality)} && \text{(59b)} \\
&\leq \frac{\alpha_q}{2L} \left(\left\| A_q^{(l)} \right\|_2 + \mathbb{E}\left[\left\| A_q^{(l)} \right\|_2 \right] \right) \qquad \text{(Jensen's inequality)} && \text{(59c)} \\
&\leq \frac{C}{2L} (8N + 8N) && \text{(59d)} \\
&= \frac{C}{2L} 16N = \frac{8CN}{L}, && \text{(59e)}
\end{aligned}$$

where $C = \sqrt{\sum_{q=1}^{Q} \alpha_q^2}$ and

$$\begin{aligned}
\boldsymbol{c}_{qk}^{(l)} &= \left[\cos\left(\mathbf{w}_{qk}^{(l)\top} \mathbf{x}_1 \right), \ldots, \cos\left(\mathbf{w}_{qk}^{(l)\top} \mathbf{x}_N \right) \right]^\top \in \mathbb{R}^N, \\
\boldsymbol{s}_{qk}^{(l)} &= \left[\sin\left(\mathbf{w}_{qk}^{(l)\top} \mathbf{x}_1 \right), \ldots, \sin\left(\mathbf{w}_{qk}^{(l)\top} \mathbf{x}_N \right) \right]^\top \in \mathbb{R}^N, \\
A_q^{(l)} &= \boldsymbol{c}_{q1}^{(l)} \boldsymbol{c}_{q1}^{(l)\top} + \boldsymbol{s}_{q1}^{(l)} \boldsymbol{s}_{q1}^{(l)\top} + \boldsymbol{c}_{q2}^{(l)} \boldsymbol{c}_{q2}^{(l)\top} + \boldsymbol{s}_{q2}^{(l)} \boldsymbol{s}_{q2}^{(l)\top} + \boldsymbol{c}_{q1}^{(l)} \boldsymbol{c}_{q2}^{(l)\top} + \boldsymbol{s}_{q1}^{(l)} \boldsymbol{s}_{q2}^{(l)\top} + \boldsymbol{c}_{q2}^{(l)} \boldsymbol{c}_{q1}^{(l)\top} + \boldsymbol{s}_{q2}^{(l)} \boldsymbol{s}_{q1}^{(l)\top}.
\end{aligned} \tag{60}$$

We are interested in bounding the spectral norm of $A_q^{(l)}$, where

$$A_q^{(l)} = \left(\mathbf{c}_{q1}^{(l)} \mathbf{c}_{q1}^{(l)\top} + \mathbf{s}_{q1}^{(l)} \mathbf{s}_{q1}^{(l)\top} + \mathbf{c}_{q2}^{(l)} \mathbf{c}_{q2}^{(l)\top} + \mathbf{s}_{q2}^{(l)} \mathbf{s}_{q2}^{(l)\top} + \mathbf{c}_{q1}^{(l)} \mathbf{c}_{q2}^{(l)\top} + \mathbf{s}_{q1}^{(l)} \mathbf{s}_{q2}^{(l)\top} + \mathbf{c}_{q2}^{(l)} \mathbf{c}_{q1}^{(l)\top} + \mathbf{s}_{q2}^{(l)} \mathbf{s}_{q1}^{(l)\top} \right).$$

- $\mathbf{c}_{qk}^{(\ell)}$ and $\mathbf{s}_{qk}^{(\ell)}$ are N-dimensional vectors, where each element is given by a cosine or sine function. Thus, we have: $\|\mathbf{c}_{qk}^{(\ell)}\|_2 \leq \sqrt{N}, \quad \|\mathbf{s}_{qk}^{(\ell)}\|_2 \leq \sqrt{N}$.
- For any vector $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$, the spectral norm of $\mathbf{a}\mathbf{b}^\top$ satisfies: $\|\mathbf{a}\mathbf{b}^\top\|_2 = \|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2$. Note that each term in $A_q^{(\ell)}$ is of the form $\mathbf{a}\mathbf{b}^\top$. Therefore, since $\|\mathbf{a}\|_2 \leq \sqrt{N}$ and $\|\mathbf{b}\|_2 \leq \sqrt{N}$ as shown earlier, we have: $\|\mathbf{a}\mathbf{b}^\top\|_2 \leq N, \quad$ for all terms in $A_q^{(\ell)}$.

Since $A_q^{(\ell)}$ is the sum of 8 such terms, we apply the subadditivity of the spectral norm::

$$\|A_q^{(\ell)}\|_2 \leq \sum_{j=1}^{8} \|\mathbf{a}_j \mathbf{b}_j^\top\|_2 \leq 8N.$$

The inequality (C.6) serves as a valid upper bound. However, it is not necessarily tight, since the spectral norm of a sum of rank-one matrices is typically less than the sum of their individual norms unless the corresponding vectors are perfectly aligned. At last, this upper bound leads directly to the inequality in Eq. (59d).

Step 3 : We first have the following bound:

$$
\begin{aligned}
&\frac{4L^2}{\alpha_q^2}\mathbb{E}\left[\left(\mathbf{E}_q^{(l)}\right)^2\right] = \mathbb{E}\left[(A_q^{(l)})^2\right] - \left(\mathbb{E}\left[A_q^{(l)}\right]\right)^2 && \text{(61a)}\\
&\preccurlyeq \mathbb{E}\left[(A_q^{(l)})^2\right] && \text{(61b)}\\
&= \mathbb{E}\left[\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}+\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}\right)\right] && \text{(61c)}\\
&= \mathbb{E}\left[\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left((\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + (\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)})\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top} + (\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right)\right] && \text{(61d)}\\
&= \sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(\mathbb{E}\left[(\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + (\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)})\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top} + (\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right) && \text{(61e)}\\
&\preccurlyeq \sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(N\mathbb{E}\left[\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}\right] + \mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right) && \text{(61f)}\\
&= 4N\mathbb{E}\left[A_q^{(l)}\right] + \sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(\mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right), && \text{(61g)}
\end{aligned}
$$

where the notation $\mathbf{A} \preccurlyeq \mathbf{B}$ denotes that $\mathbf{B} - \mathbf{A}$ is a PSD matrix. To ensure the validity of the inequality in Eq. (61b), it is necessary to show that $\left(\mathbb{E}[A_q^{(l)}]\right)^2$ is a positive semi-definite (PSD) matrix.

Since $A_q^{(\ell)}$ is a real symmetric matrix, its expectation $\mathbb{E}[A_q^{(\ell)}]$ is also real symmetric. Therefore, we can apply the following result:

Lemma 4 (Square of a Real Symmetric Matrix is PSD)**.** *Let $M \in \mathbb{R}^{n\times n}$ be a real symmetric matrix. Then its square $M^2 := MM$ is positive semi-definite, i.e., $M^2 \succeq 0$.*

Proof. Since M is symmetric, it admits an eigendecomposition of the form $M = U\Lambda U^\top$, where $U \in \mathbb{R}^{n\times n}$ is an orthogonal matrix and $\Lambda \in \mathbb{R}^{n\times n}$ is a real diagonal matrix of eigenvalues. Then,

$$
M^2 = MM = U\Lambda U^\top U\Lambda U^\top = U\Lambda^2 U^\top,
$$

where Λ^2 is the diagonal matrix of squared eigenvalues of M. Since the square of any real number is non-negative, all eigenvalues of M^2 are non-negative. Therefore, M^2 is PSD. □

Applying Lemma 4 to $\mathbb{E}[A_q^{(l)}]$, we conclude that $\left(\mathbb{E}[A_q^{(l)}]\right)^2$ is PSD. This confirms that the inequality in Eq. (61b) holds.

The inequality in Eq. (61f) holds because

$$
\begin{aligned}
&N\mathbb{E}\left[\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + \boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}\right] - \mathbb{E}\left[(\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + (\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)})\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}\right]\\
&= \mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{s}_{qk_3}^{(l)})\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top} + (\boldsymbol{c}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top}\right] \qquad \text{(62)}\\
&\left(\text{due to } \boldsymbol{c}_{qk_2}^{(l)}\boldsymbol{c}_{qk_3}^{(l)\top} + \boldsymbol{s}_{qk_2}^{(l)}\boldsymbol{s}_{qk_3}^{(l)\top} = \sum_{j=1}^{N}(\cos\left(\mathbf{w}_{qk_2}^{(l)\top} - \mathbf{w}_{qk_3}^{(l)\top}\right)\mathbf{x}_j) \preccurlyeq N\right)
\end{aligned}
$$

is a PSD matrix.

Then we are able to bound the variance, $\left\|\sum_{q=1}^{Q}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_q^{(l)})^2]\right\|_2$, as

$$
\begin{aligned}
&\left\|\sum_{q=1}^{Q}\sum_{l=1}^{L/2}\mathbb{E}[(\mathbf{E}_q^{(l)})^2]\right\|_2 \\
&\leq \left\|\sum_{q=1}^{Q}\sum_{l=1}^{L/2}\frac{\alpha_q^2}{4L^2}\left(4N\mathbb{E}\left[A_q^{(l)}\right]+\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(\mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right)\right)\right\|_2 \\
&\leq \frac{2C}{L}\left\|\sum_{q=1}^{Q}\alpha_q\left(\frac{N}{4}\mathbb{E}\left[A_q^{(l)}\right]+\frac{1}{16}\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\left(\mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right)\right)\right\|_2 \\
&\leq \frac{2C}{L}\left(N\left\|\mathbf{K}_{\mathrm{ngsm}}\right\|_2+\frac{1}{16}\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\sum_{q=1}^{Q}\alpha_q\left\|\mathbb{E}\left[(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right]\right\|_2\right) \quad \text{(triangle inequality)} \\
&\leq \frac{2C}{L}\left(N\left\|\mathbf{K}_{\mathrm{ngsm}}\right\|_2+\frac{1}{16}\sum_{k_1=1}^{2}\sum_{k_2=1}^{2}\sum_{k_3=1}^{2}\sum_{k_4=1}^{2}\sum_{q=1}^{Q}\alpha_q\mathbb{E}\left[\left\|(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right\|_2\right]\right) \quad \text{(Jensen's inequality)} \\
&\leq \frac{2C}{L}\left(N\left\|\mathbf{K}_{\mathrm{ngsm}}\right\|_2+\frac{1}{16}\sum_{k_1=1}^{2}\sum_{k_4=1}^{2}4*\frac{N}{2}\sum_{q=1}^{Q}\alpha_q\mathbb{E}\left[\left\|(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right\|_2\right]\right) \qquad \left(|(\boldsymbol{s}_{qk_2}^{(l)\top}\boldsymbol{c}_{qk_3}^{(l)})|\leq\frac{N}{2}\right) \\
&\leq \frac{2CN}{L}\left(\left\|\mathbf{K}_{\mathrm{ngsm}}\right\|_2+\frac{1}{16}*\frac{N}{2}*16C\sqrt{Q}\right),
\end{aligned}
\tag{63}
$$

where the last inequality is because that

$$
\mathbb{E}\left[\left\|(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\right\|_2\right]=\sup_{\|\boldsymbol{v}\|_2^2=1}\mathbb{E}\left[\left\|\boldsymbol{v}^\top(\boldsymbol{s}_{qk_1}^{(l)}\boldsymbol{c}_{qk_4}^{(l)\top}+\boldsymbol{c}_{qk_1}^{(l)}\boldsymbol{s}_{qk_4}^{(l)\top})\boldsymbol{v}\right\|_2\right]\leq N, \tag{64}
$$

and $\sum_{q=1}^{Q}\alpha_q\leq C\sqrt{Q}$ by the Cauchy–Schwarz inequality.

Step 4 : We can now apply the derived upper bounds, given by Eqs. (59) and (63), to H and $v(\mathbf{E})$ in Lemma 3,

$$
P\left(\left\|\hat{\mathbf{K}}_{\mathrm{ngsm}}-\mathbf{K}_{\mathrm{ngsm}}\right\|_2\geq\epsilon\right)\leq N\exp\left(\frac{-3\epsilon^2L}{2NC\left(6\left\|\mathbf{K}_{\mathrm{ngsm}}\right\|_2+3NC\sqrt{Q}+8\epsilon\right)}\right), \tag{65}
$$

which completes the proof of Theorem 4.

□

D EXPERIMENTAL DETAILS

D.1 DATASET DESCRIPTION

This section presents a detailed overview of the datasets used in our experiments, covering the generation process of synthetic data, the description of real-world data, and the applied preprocessing steps.

D.1.1 Synthetic Data

The datasets are generated using a 2-view MV-GPLVM with an S-shaped latent variable, employing two different kernel configurations: (1) both views use the RBF kernel, and (2) one view uses the RBF kernel while the other uses the Gibbs kernel. Detailed descriptions of the kernels are provided below.

- **RBF Kernel**: The kernel function is expressed as:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \ell_o \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\ell_l^2}\right)$$

where $\ell_o = 1$ denotes the outputscale, and $\ell_l = 1$ represents the lengthscale.

- **Gibbs Kernel**: As a non-stationary kernel, the kernel function is formulated as:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\frac{2\ell_{\mathbf{x}_1}\ell_{\mathbf{x}_2}}{\ell_{\mathbf{x}_1}^2 + \ell_{\mathbf{x}_2}^2}} \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{\ell_{\mathbf{x}_1}^2 + \ell_{\mathbf{x}_2}^2}\right)$$

In this context, $\ell_{\mathbf{x}}$ is dynamic length scales derived from the positions of the input point $\mathbf{x}$, specifically defined as:

$$\ell_{\mathbf{x}} = \exp\left(-0.5 \cdot \|\mathbf{x}\|\right).$$

D.1.2 Real-World Data

Table 5: Description of real-world datasets.

DATASET	# SAMPLES (N)	# DIMENSIONS (M)	# LABELS
BRIDGES	2,000	2	2
CIFAR	60,000	32 × 32	10
R-CIFAR	2,000	20 × 20	5
MNIST	70,000	28 × 28	10
R-MNIST	1,000	28 × 28	10
NEWSGROUPS	20,000	19	3
YALE	165	32 × 32	15
CHANNEL	20,000	64	10
BRENDAN	2,000	20 × 28	-

The detailed preprocessing steps and descriptions of the real-world datasets are provided below. For convenience, key information about the datasets is summarized in Table 5.

- 1) <u>BRIDGES:</u> This dataset is a collection of data documenting the daily bicycle counts crossing four East River bridges in New York City[1] (Brooklyn, Williamsburg, Manhattan, and Queensboro). We classify the data into weekdays and weekends, treating these as binary labels. This classification aims to examine the variations in bicycle counts on weekdays versus weekends, exploring whether significant differences exist in the traffic patterns between the two.
- 2) <u>CIFAR:</u> This dataset comprises 60,000 color images with a resolution of 32×32 pixels, categorized into 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. We convert the color image into grayscale image for training.
- 3) <u>R-CIFAR:</u> We sample this dataset from the CIFAR dataset, specifically selecting five categories: airplane, automobile, bird, cat, and deer. For each category, 400 images are sampled, and each 32×32 pixel image is converted into a 20×20 pixel image.

[1] `https://data.cityofnewyork.us/Transportation/Bicycle-Counts-for-East-River-Bridges/gua4-p9wg`

- 4) MNIST: It is a classic handwritten digit recognition dataset. It consists of 70,000 grayscale images with a resolution of 28×28 pixels, divided into 60,000 training samples and 10,000 testing samples. Each image represents a handwritten digit ranging from 0 to 9.
- 5) R-MNIST: We select 1,000 randomly handwritten digit images from the classic MNIST dataset.
- 6) NEWSGROUPS: It is a dataset for text classification, containing articles from multiple newsgroups[2]. We restrict the vocabulary to words that appear within a document frequency range of 10% to 90%. For our analysis, we specifically select text from three classes: comp.sys.mac.hardware, sci.med, and alt.atheism.
- 7) YALE: The Yale Faces Dataset[3] consists of face images from 15 different individuals, captured under various lighting conditions, facial expressions, and viewing angles.
- 8) BRENDAN: The dataset contains 2,000 photos of Brendan's face[4].

D.1.3 Channel Data

To evaluate model performance under realistic wireless environments, we simulate channel data using the QUAsi Deterministic RadIo channel GenerAtor (QuaDRiGa)[5], a widely-used geometry-based stochastic channel simulator. The parameter settings used in our experiments are listed in Table 6.

The simulation scenario emulates a typical urban communication setting, where the user equipment (UE) moves at a constant speed of 30 km/h. QuaDRiGa accounts for both the speed and the direction of movement, which significantly influence the channel behavior. Specifically, the relative motion between the UE and the base station (BS) introduces Doppler shifts and time-varying path loss. As the UE moves through the environment, the geometry between the transmitter, receiver, and scatterers changes over time, leading to dynamic variations in multipath components such as delays, angles, and Doppler frequencies.

It is worth noting that no explicit additive noise (e.g., Gaussian noise) is introduced in the simulation. Instead, the randomness in the channel arises naturally from QuaDRiGa's stochastic modeling of multipath fading, spatial non-stationarity, and time dynamics. These controlled yet realistic variations ensure consistency across different simulation runs while providing sufficient complexity to evaluate the robustness of the proposed model.

D.2 BENCHMARK METHODS

We provide a description of the benchmark methods as follows:

- 1) PCA: A classical linear dimensionality reduction method that projects data to orthogonal components maximizing variance.
- 2) LDA: A supervised linear method that finds projections maximizing class separability.
- 3) ISOMAP: A nonlinear manifold learning method that preserves geodesic distances between data points.
- 4) HPF: A hierarchical poisson factorization model for probabilistic matrix factorization, commonly used in recommendation systems.
- 5) BGPLVM: A bayesian formulation of GPLVM that places a prior over latent variables and infers their posterior using variational inference.
- 6) GPLVM-SVI: A scalable GPLVM variant using stochastic variational inference to handle large datasets.
- 7) VAE: A deep generative model that learns latent representations via amortized variational inference.
- 8) NBVAE: A probabilistic model for sparse, overdispersed count data, combining a Gaussian process prior with a negative binomial likelihood for flexible, non-linear modeling.
- 9) DCA: The deep count autoencoder is designed to denoise single-cell RNA sequencing data by accounting for count distribution, overdispersion, and sparsity, using a zero-inflated negative binomial noise model.

[2] http://qwone.com/~jason/20Newsgroups/
[3] http://vision.ucsd.edu/content/yale-face-database
[4] https://cs.nyu.edu/~roweis/data/frey_rawface.mat
[5] https://quadriga-channel-model.de

Table 6: Parameters settings of QUADRIGA.

PARAMETER DESCRIPTION	VALUES
Overall Setup	
# samples	1000
# user equipment (UE)	10
# receive antennas	1
Moving speed (km/h)	30
Proportion of indoor UEs	1
Time sampling interval (seconds)	5e-3
Total duration of sampling (seconds)	5
Channel type	3GPP_3D_UMa
Channel Configuration	
Center frequency (Hz)	1.84e9
Use random initial phase	False
Use geometric polarization	False
Use spherical waves	False
Show progress bars	False
Base Station (BS) Antenna Configuration	
# vertical elements per antenna	4
# horizontal elements per antenna	1
# rows in the antenna array	2
# columns in the antenna array	8
Electrical tilt angle (degrees)	7
# carriers	1
# transmit antennas	32
UE Antenna Configuration	
UE antenna array	omni
# subcarriers	1
Subcarrier spacing (Hz)	1e6
# loops (simulations)	1
Minimum UE distance from BS (meters)	35
Maximum UE distance from BS (meters)	300
Layout Configuration	
# base stations	1
Base station position (x, y, z) (meters)	(0, 0, 30)
UE movement path length (meters)	41.667

- 10) CVQ-VAE: The clustering vector quantised variational aautoencoder is an enhanced VQ-VAE model designed to prevent codebook collapse by utilizing an online clustered codebook.
- 11) RFLVM: GPLVM extension using random feature approximation for scalable inference.
- 12) DGPLVM: A deep extension of GPLVM that stacks multiple GP layers to learn hierarchical nonlinear mappings from latent space, typically trained via variational inference.
- 13) ARFLVM: An advised GPLVM using kernel learning and latent regularization for flexible representation learning.
- 14) MVAE: Multi-modal VAE using a product-of-experts (PoE) to infer shared latent representations from multiple modalities.
- 15) MMVAE: Mixture-of-experts (MoE) based VAE that balances shared and private representations for multi-modal learning.

Table 7 provides the references and implementation details for each benchmark method, aiming to enhance reproducibility.

D.3 HYPERPARAMETER SETTINGS

Figure 9 depicts the latent manifold learning outcomes of NG-RFLVM for varying values of Q and L on synthetic single-view data. Figure 5 shows a heatmap of R^2 scores, quantifying the similarity between the learned and ground truth latent variables, with values closer to 1 indicating better alignment. Figure 6 displays a heatmap of wall-time across varying values of Q and L. These results demonstrate that larger values of Q and L typically improve model performance, albeit at the cost of higher computational complexity. To balance computational efficiency with latent representation quality, we select $Q = 2$ and $L = 50$. The default hyperparameter settings are summarized in Table 8.

Table 7: Descriptions of benchmark methods.

METHOD	REFERENCE	IMPLEMENTATION CODE
PCA	Wold et al. [1987]	Using the `scikit-learn` library [Buitinck et al., 2013].
LDA	Blei et al. [2003]	Using the `scikit-learn` library [Buitinck et al., 2013].
ISOMAP	Balasubramanian and Schwartz [2002]	Using the `scikit-learn` library [Buitinck et al., 2013].
HPF	Gopalan et al. [2015]	`https://github.com/david-cortes/hpfrec`
BGPLVM	Titsias and Lawrence [2010]	`https://github.com/SheffieldML/GPy`
GPLVM-SVI	Lalchand et al. [2022]	`https://github.com/vr308/Generalised-GPLVM`
VAE	Kingma and Welling [2013]	`https://github.com/pytorch/examples/blob/main/vae/main.py`
NBVAE	Zhao et al. [2020]	`https://github.com/ethanheathcote/NBVAE`
DCA	Eraslan et al. [2019]	`https://github.com/theislab/dca`
CVQ-VAE	Zheng and Vedaldi [2023]	`https://github.com/lyndonzheng/CVQ-VAE`
RFLVM	Zhang et al. [2023], Gundersen et al. [2021]	`https://github.com/gwgundersen/rflvm`
DGPLVM	Salimbeni and Deisenroth [2017]	`https://github.com/UCL-SML/Doubly-Stochastic-DGP`
ARFLVM	Li et al. [2024]	`https://github.com/zhidilin/advisedGPLVM`
MVAE	Wu and Goodman [2018]	`https://github.com/mhw32/multimodal-vae-public`
MMVAE	Shi et al. [2019], Mao et al. [2023]	`https://github.com/OpenNLPLab/MMVAE-AVS`

Table 8: Default Hyperparameter Settings.

PARAMETER DESCRIPTION	VALUES
NG-SM Kernel Setup	
# Mixture densities (Q)	2
Dim. of random feature ($L/2$)	50
Dim. of latent space (D)	2
Optimizer Setup (Adam)	
Learning rate	0.01
Beta	(0.9, 0.99)
# Iterations	10000

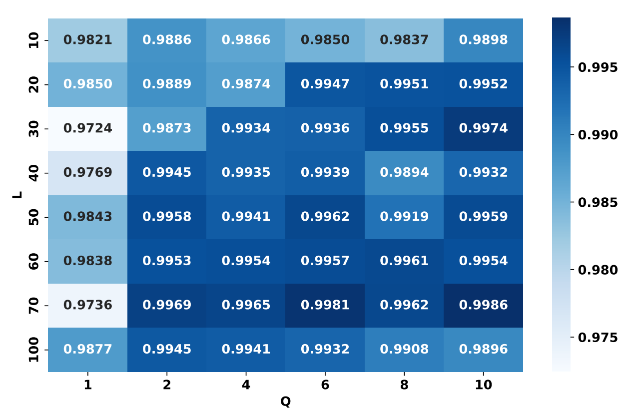


Figure 5: Heatmap of R^2 in latent manifold learning.

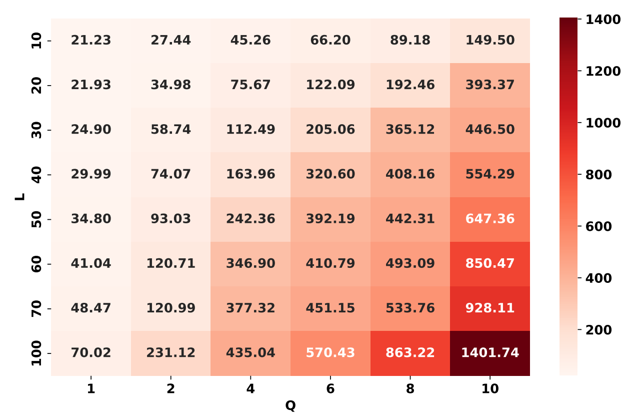


Figure 6: Heatmap of wall-time (seconds) in latent manifold learning.

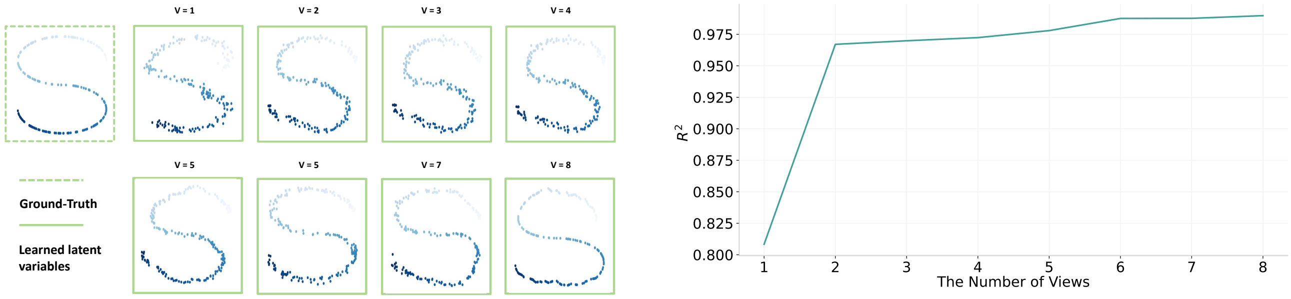
D.4 ADDITIONAL EXPERIMENTS ON MULTI-VIEW DATASETS

To explore more general scenarios, we conducted experiments on synthetic datasets and multi-view MNIST, both containing a large number of views.

D.4.1 Multi-View Synthetic Data

As the number of views increases, Figures 7a and 7b present the unified latent representations generated by our method and the corresponding R^2 scores, respectively. These results show that with more views, the unified latent representations learned by our model progressively align more closely with the ground truth.

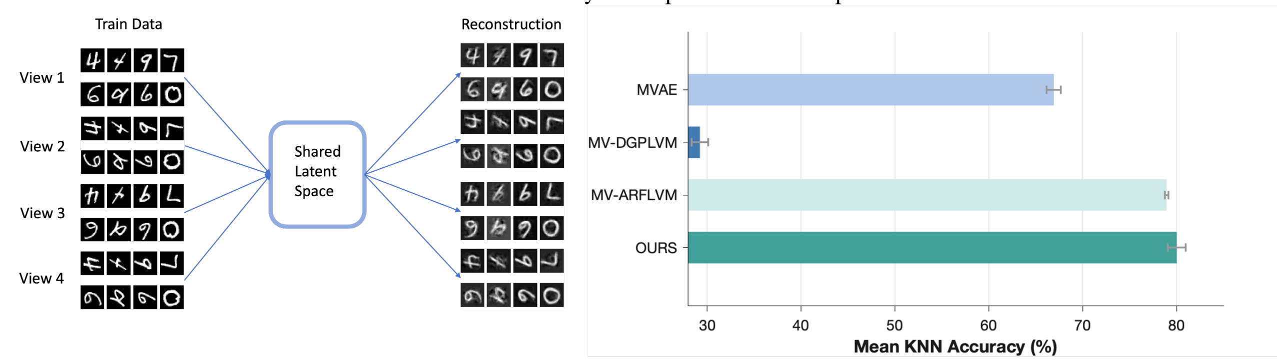
D.4.2 Multi-View MNIST



(a) Latent manifold learning results with multiple views. **(b)** R^2 against the number of views.

Figure 7: Comparison of latent manifold learning and R^2 against the number of views.

Figure 8: (**Left**) MV-MNIST reconstruction task. (**Right**) Classification accuracy (%) evaluated using KNN classifier with five-fold cross-validation. Mean and standard deviation of the accuracy is computed over five experiments.



We generated a four-view dataset derived from the MNIST dataset using a rotation operation, as illustrated on the left-hand side of Figure 8, alongside reconstruction results obtained with our method. The right-hand side of Figure 8 displays the KNN accuracy evaluated using the latent variables learned by various methods on MV-MNIST. These results highlight that our approach not only achieves superior performance in downstream classification tasks but also effectively reconstructs data form each view through the shared latent space.

D.5 MISSING DATA IMPUTATION

In this section, we evaluate our model's capability for missing data imputation using the single-view datasets MNIST and BRENDAN. We randomly set various proportions of the observed data $\mathbf{Y}$ to zero (denoted as $\mathbf{Y}_{\text{obs}}$), ranging from 0% to 60%. Using the incomplete datasets $\mathbf{Y}_{\text{obs}}$, our model estimates the underlying latent variable $\mathbf{X}$, and the missing values are imputed as $\hat{\mathbf{Y}}_{\text{miss}} = \mathbb{E}[\mathbf{Y}_{\text{miss}} \mid \mathbf{X}, \mathbf{Y}_{\text{obs}}]$. Figures 10 and 11 illustrate the reconstruction tasks on the MNIST and BRENDAN datasets with varying proportions of missing values, showing superior ability of our model to restore missing pixels.[6]

[6]In the future, we particularly interested in expressing a factorized latent space where each view is paired with an additional private space, alongside a shared space to capture unaligned variations across different views [Damianou et al., 2012, 2021].

E EXTENDED RELATED WORK

E.1 MULTI-VIEW VARIATIONAL AUTOENCODERS

Multi-view variational autoencoders (MV-VAEs) incorporate various view-specific variational autoencoders (VAEs) to address multi-view representation learning[Mao et al., 2023, Xu et al., 2021, Wu and Goodman, 2018, Shi et al., 2019, Palumbo et al., 2023, Sutter et al., 2020, Hwang et al., 2021]. A class of MV-VAEs fuse multi-view information through different posterior combination strategies [Wu and Goodman, 2018, Shi et al., 2019, Palumbo et al., 2023]. For instance, MVAE [Wu and Goodman, 2018] employs a product-of-experts (PoE) approach, enabling scalable inference but often resulting in overconfident posterior estimates. To mitigate this, MMVAE [Shi et al., 2019] adopts a mixture-of-experts (MoE) formulation, which better captures unimodal distributions but tends to underutilize shared information across modalities. MMVAE+ [Palumbo et al., 2023] further improves generative performance by disentangling the latent space into shared and private components, thereby promoting both cross-modal consistency and modality-specific flexibility.

Another line of work advances multi-view representation learning by modifying the training objective. For example, mmJSD [Sutter et al., 2020] replaces the standard ELBO with a Jensen-Shannon divergence-based objective to improve training stability. MVT-CVAE [Hwang et al., 2021] further introduces an information-theoretic formulation that minimizes total correlation in the latent space, thereby encouraging disentanglement and enabling robust generalization to missing views.

Despite these advances, VAE-based models often face posterior collapse [Wang et al., 2021], where the latent variables fail to capture informative structure. One major cause is the overfitting of the decoder network [Bowman et al., 2016, Sønderby et al., 2016]. Consequently, MV-VAEs may struggle to learn informative unified latent representations, particularly in data-scarce regimes, where overfitting and posterior collapse are more pronounced.

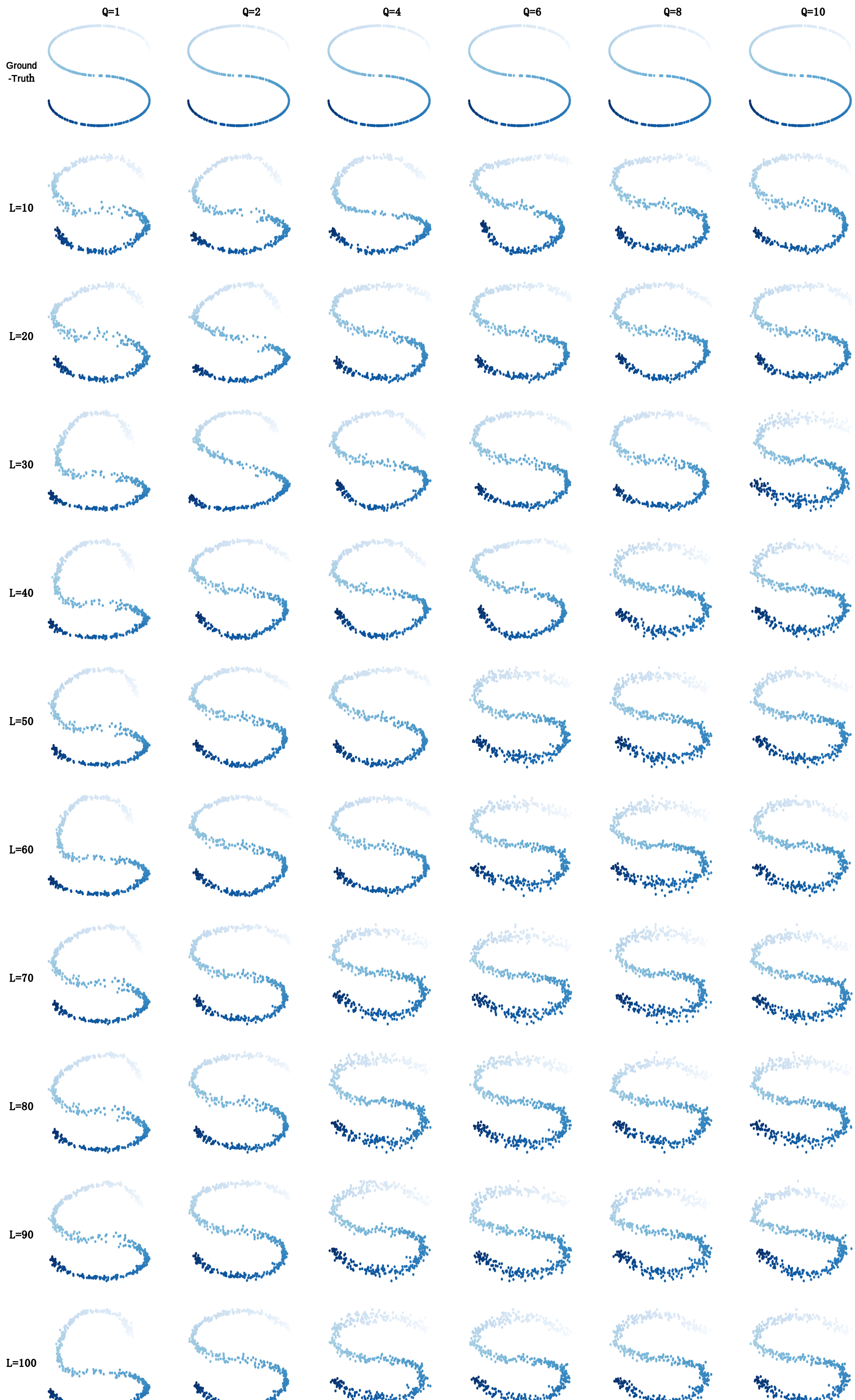
E.2 MULTI-VIEW CLUSTERING

Recent works in multi-view clustering exploit structural information to enhance clustering performance. PANDA constructs topological manifolds from view-specific spatial graphs and encodes them into a low-rank tensor using a refined sigmoid function to preserve significant singular values [Gu et al., 2024b], thereby capturing both intra-view topological structures and inter-view correlations. NOODLE disentangles self-representations into consistent and inconsistent components [Gu et al., 2024a], applies cross-view sparsity to isolate discrepancies, and stacks the purified representations into a tensor to jointly model high-order dependencies and shared structure.

While the above methods represent clustering-oriented approaches, our work focuses on a different subarea: multi-view representation learning (MVRL), which involves distinct objectives and modeling strategies. Specifically, PANDA and NOODLE aim to construct $N \times N$ pairwise affinity matrices for clustering, whereas we seek to learn unified low-dimensional latent representations from multi-view observations.

Although our focus differs, the ideas in PANDA and NOODLE—such as capturing structural relationships across views and leveraging tensor-based modeling—offer valuable insights for designing better latent variable models. We leave the exploration of these directions to future work and concentrate in this paper on MVRL.

Figure 9: Latent manifold learning results with different Q and L.



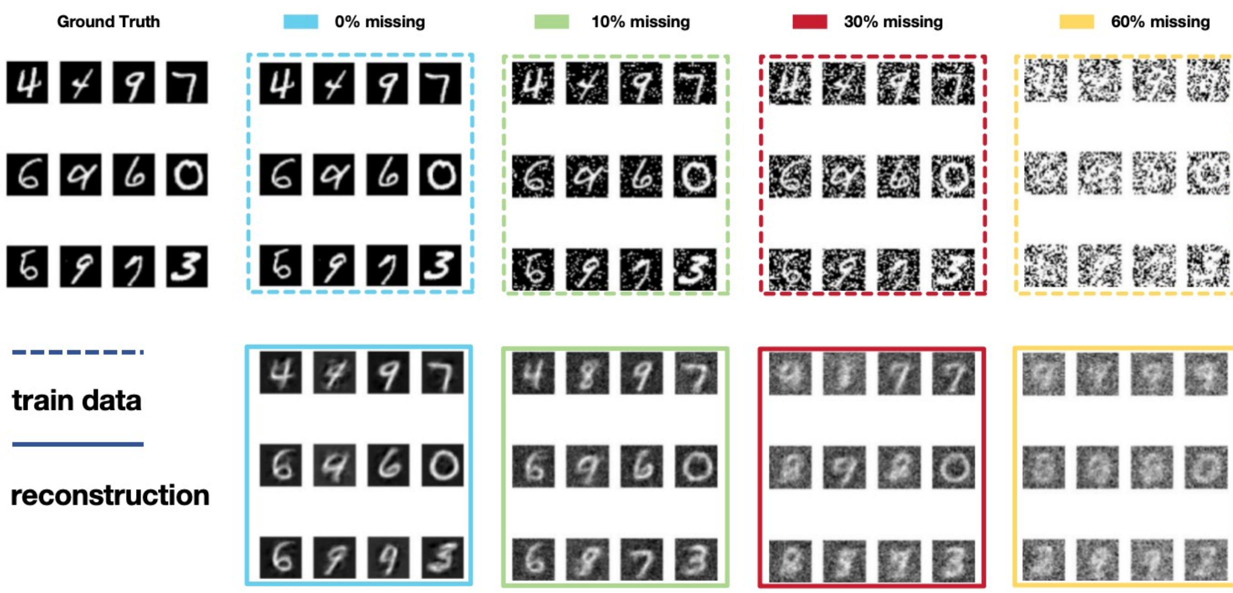


Figure 10: MNIST reconstruction task.

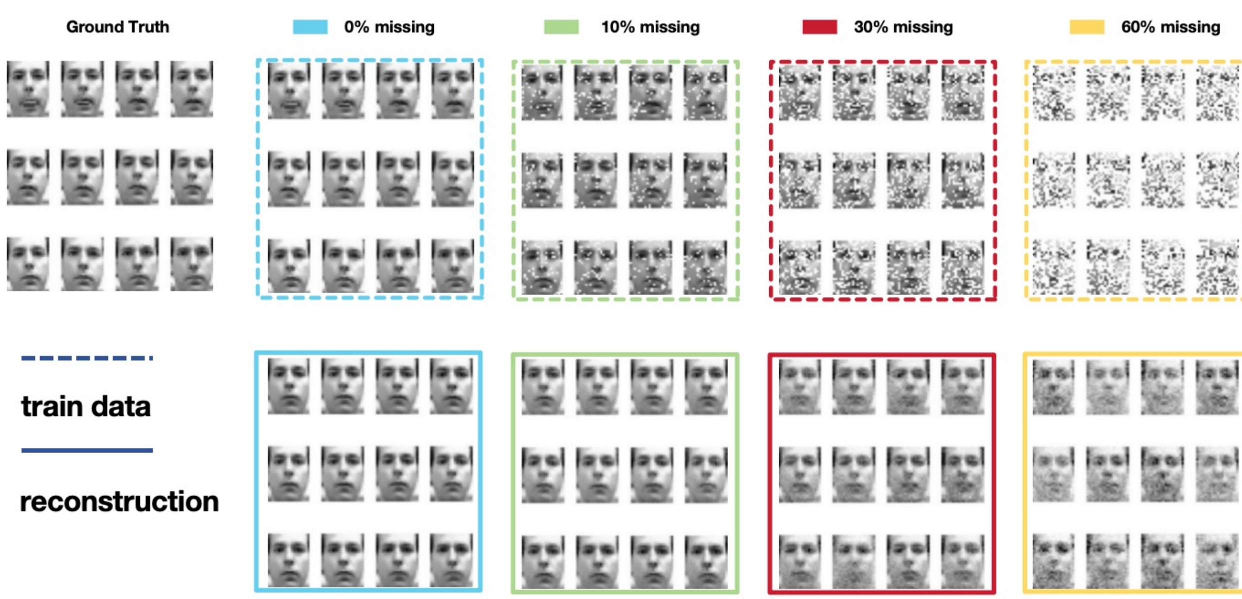


Figure 11: BRENDAN reconstruction task.